RECONSTRUCTION FROM LIMITED PROJECTIONS

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

By

AMITABHA SANYAL

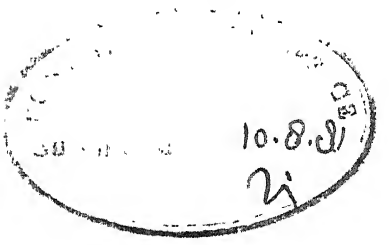
to the

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

AUGUST, 1981

EE-1981-M- SAN-REC



CERTIFICATE

This is to certify that the thesis entitled "Reconstruction from Limited Projections' is a record of the work carried out under our supervision and that it has not been submitted elsewhere for a degree.

(S.K. MULLICK)
Professor
Dept. of Electrical Engg.
Indian Institute of Tech.
KANPUR

(R.K.S. RATHORE)
Assistant Professor
Department of Mathematics
Indian Institute of Tech.
KANPUR

ACKNOWLEDGEMENT

With a deep sense of gratitude, I wish to acknowledge my gratefulness to my thesis supervisors, Dr. S.K. Mullick and Dr. R.K.S. Rathore for their guidance, invaluable suggestions and encouragement throughout the course of this study.

I also thank Shri Mrinal Patra for his helps and suggestions.

Finally, thanks are due to Shri K.N. Tewari for his neat and efficient typing.

AMITABHA SANYAL

CONTENTS

CHAPTER 1

THE PROBLEM

The need for estimating the cross section of a object from its projections arises in a large number of areas. The most well known application is X-ray computerized tomography, in which X-ray scans at different angles are combined to reconstruct a cross sectional image which represents the details of the scanned structure. The performance of computer axial tomography in pinpointing tumors hemorrhages, blood clots etc. was so dramatic as to revolutionize radiology [10,24]. A few specific applications are mentioned below.

(1) Noninvasive angiograms: One of the traditional brain examinations, the angiogram, has the purpose of visualizing the arteries in the head, as well as certain tumors. The usual method of doing this is to inject iodine contrast dye into the cartoid artery in the neck. The dye goes directly to the head and then to the body. While the full amount is in the head an X-ray is taken. The direct arterial injection is painful and dangerous, always calling for hospitalization, and often producing grave complications. On the other hand, a slow intravenous injection of dye is considered relatively painless and harmless. But the concentration in this case is not sufficient for usual X-ray photography. However these concentrations are enough to be detectable in cross sectional reconstructions.

(2) Even when a mass is known to be present in the breast, it remains a difficult problem to determine without a biopsy whether the mass is cancer or a fibrocystic lesion. Since the fibrocystic lesion is more dense than the fatty breast tissue, while the cancer is still more dense, a reconstruction of the cross section of the breast can be achieved by simply examining the densitometer readings along a line through the lesion.

Apart from medical applications, there are other fields too, where reconstruction tomography can be used extensively. In radio astronomy the two dimensional brightness distribution of a source can be estimated from fan beam telescopic scans. The problem can be extended to multidimensional fields too. In electron microscopy the three dimensional structure of biological specimens can be estimated from their two dimensional electron micrographs taken at different tilt angles.

The cross sectional reconstruction can also be used to obtain three dimensional information by stacking successive transverse sections [30]. Real objects exist in the three dimensional space R^3. However, a solution to the two dimensional problem provides immediately a solution to the three dimensional problem by recovering all two dimensional sections orthogonal to a fixed line. The sections are then stacked together to obtain a three dimensional reconstruction. At present almost all three dimensional reconstructions are being obtained in this way for reasons of computational simplicity.

Another potential application is in picture transmission [19]. In transmitting pictures, one can increase the rate of transmission of information with only moderate loss of accuracy by transmitting only the projections instead of the actual picture. Consider a 256 x 256 picture. It has been seen that by reducing it to a coarser 32 x 32 picture and sending only 32 x 16 projections, the picture can be adequately reconstructed. The compression of data in this case is 256 x 256:32 x 16 = 128:1 Of course the accuracy can be increased by increasing the amount of information. But the transmitted data may be corrupted with additive noise and the reconstruction method should take into account this fact.

In general, two types of geometrical arrangements are usually employed for producing the projections of a multi-dimensional field. These are based on illuminating the source by either parallel beams or fan beams of radiation (be they heat, sound, X-ray or electromagnetic). Fig.1 illustrates a parallel beam geometry. A set of collimated sources illuminates the object and an array of sensors gather the projection data. By shifting the assembly of source and sensors through small angles, a large amount of projection data is gathered. In the case of X-ray tomography, the sources are X-ray guns and the sensors are low noise photo-detectors, which detect the amount of X-ray incident on them and the absorption in the path.

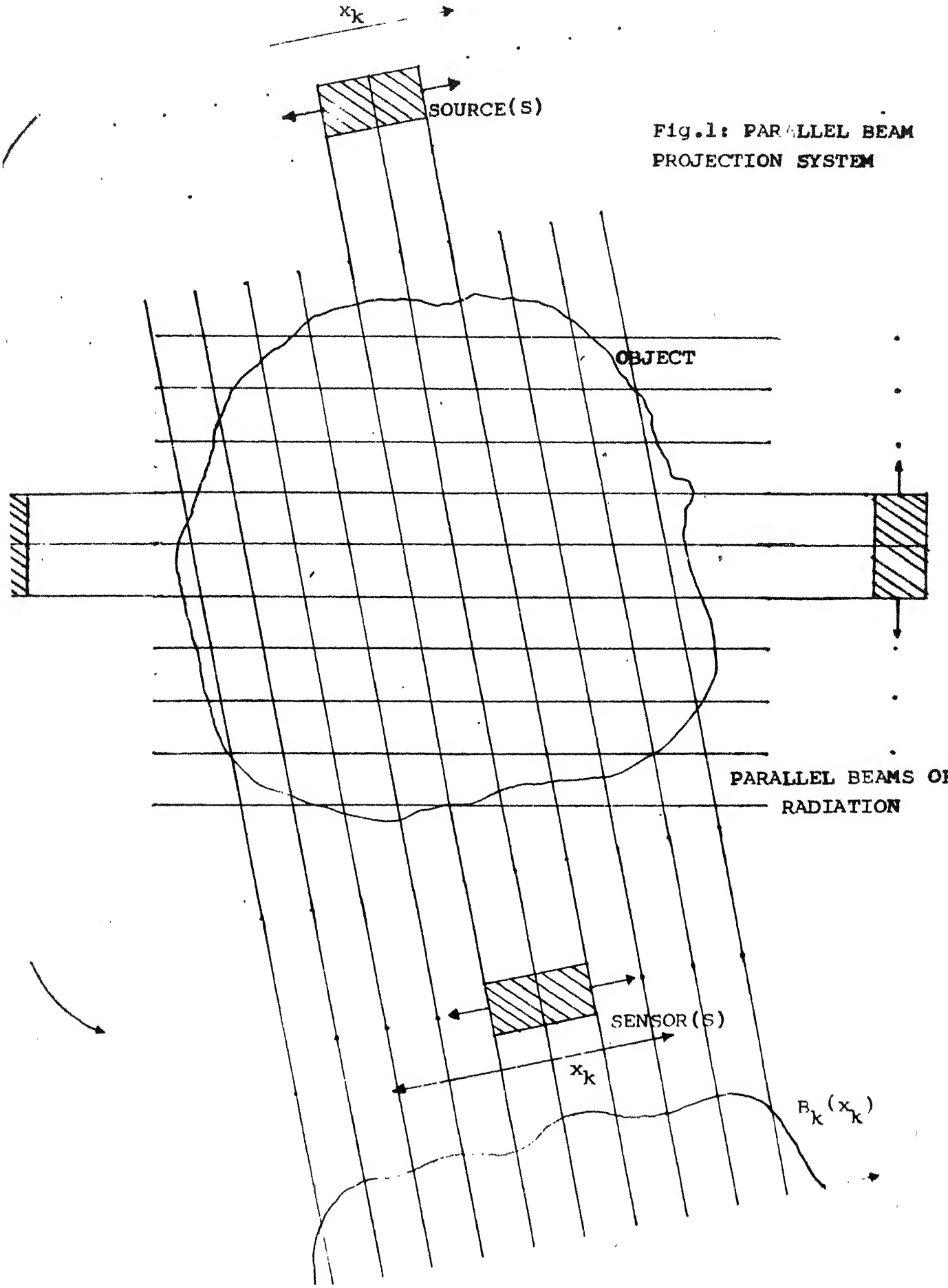


Fig.1: PARALLEL BEAM PROJECTION SYSTEM

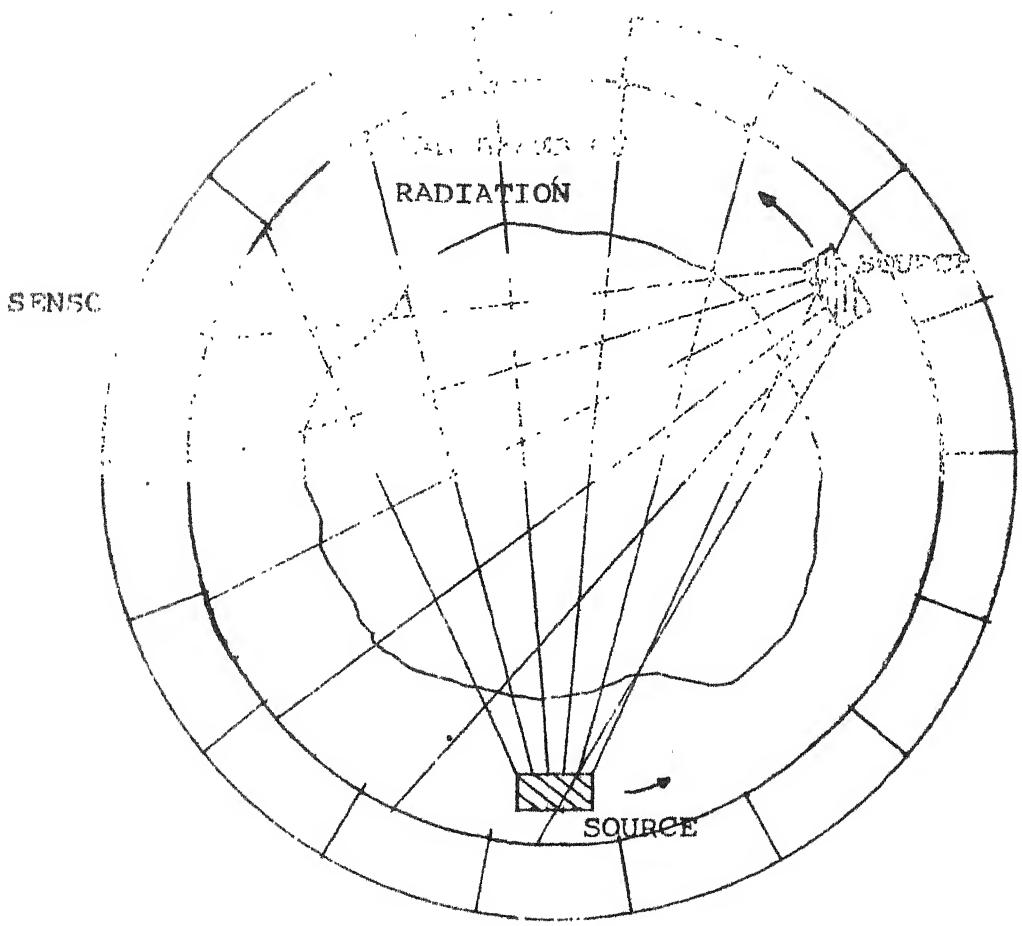


Fig.2 FAN BEAM PROJECTION SYSTEM.

Parallel beam systems, specially in X-ray tomography, require very high precision moving parts to ensure correct registration of each projection at different angles. To minimise on rotating parts and to speed up the data collection, fan beam systems have been recently introduced(Fig.2).These consist of a single rotating source which produces a fan beam of incident radiation. The radiation is collected after transmission through an object by a set of fixed sensors located on a circle. This is done for several angles and the projection data is gathered. In radio astronomy the scanning beam is essentially a fan beam.

Mathematically,the reconstruction problem may be stated as follows [25]:

(a) For parallel beam geometry (Fig.3): Given a set of N projections of a two dimensional field or object $[A(x,y)]$ at angles $[\theta_k,\ k = 1,2,\ldots,\ N]$ with respect to the reference system of coordinates, i.e.

$$B_k(x_k)dx_k = \int_S A(x,y)dy_k\ dx_k$$

where A represents the two dimensional field intensity at any point (x,y),

y_k is orthogonal to x_k,

S represents the strip of integration of A(x,y) taken at the projection angle θ_k and location x_k and

$$x_k = x\cos\theta_k + y\sin\theta_k,$$

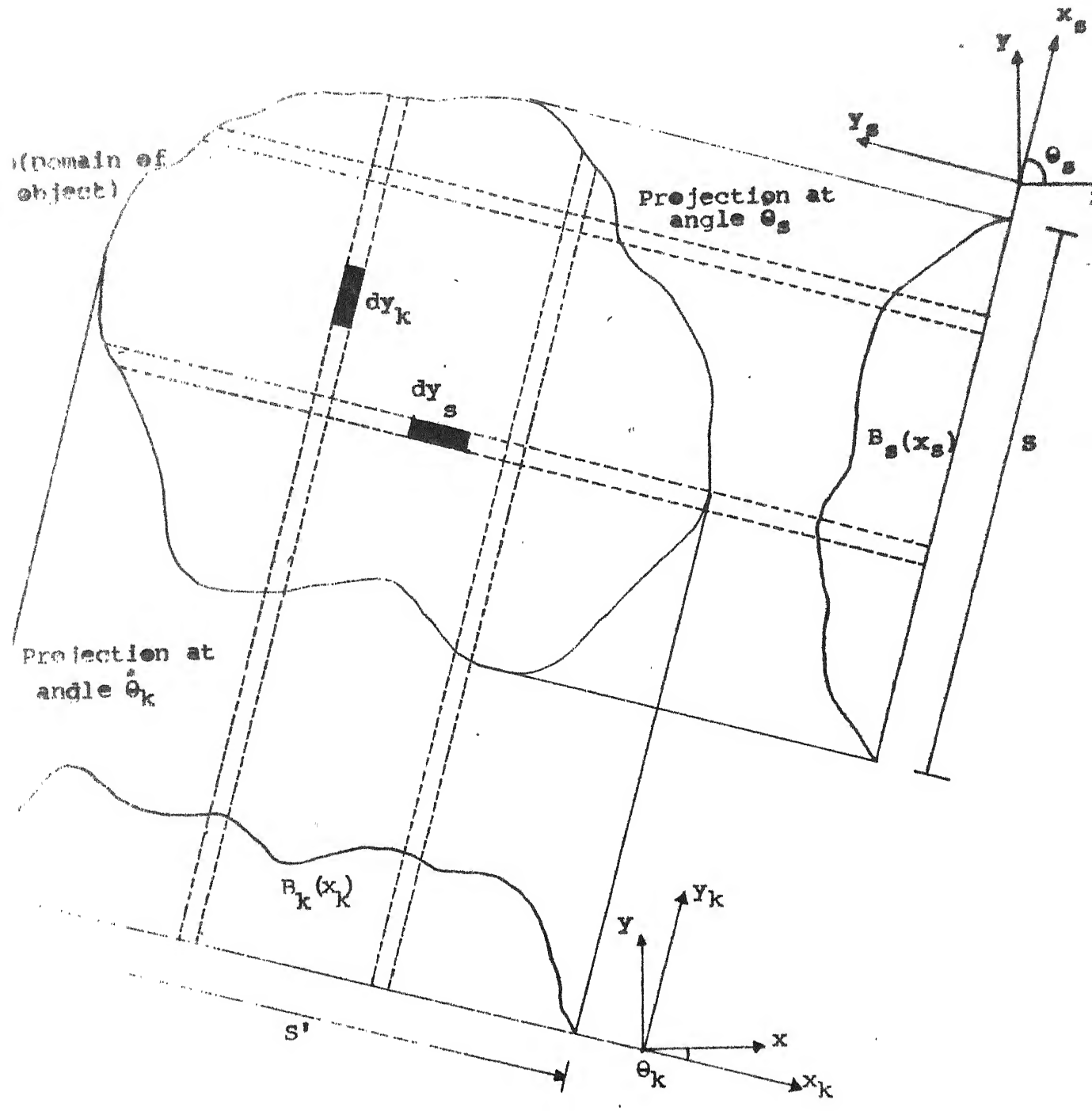


Fig.3 Acquisition of parallel beam projections.

$$y_k = -x\sin\theta_k + y\cos\theta_k$$

are the rotated Cartesian coordinates and $B_k(x_k)$ is a one-dimensional function with respect to x_k. The objective is to obtain 'good' estimates of $[A(x,y)]$ from observations of $[B_k(x_k) \forall\ x_k,\ k]$ by appropriate signal processing.

(b) Fan beam geometry (Fig.4): The fan beam projection with the source at an angle γ with respect to the reference system of coordinates is given by

$$B_\gamma(\theta_k) = \int_{l_1(\theta_k)}^{l_2(\theta_k)} A(x,y)\, dr_k$$

where

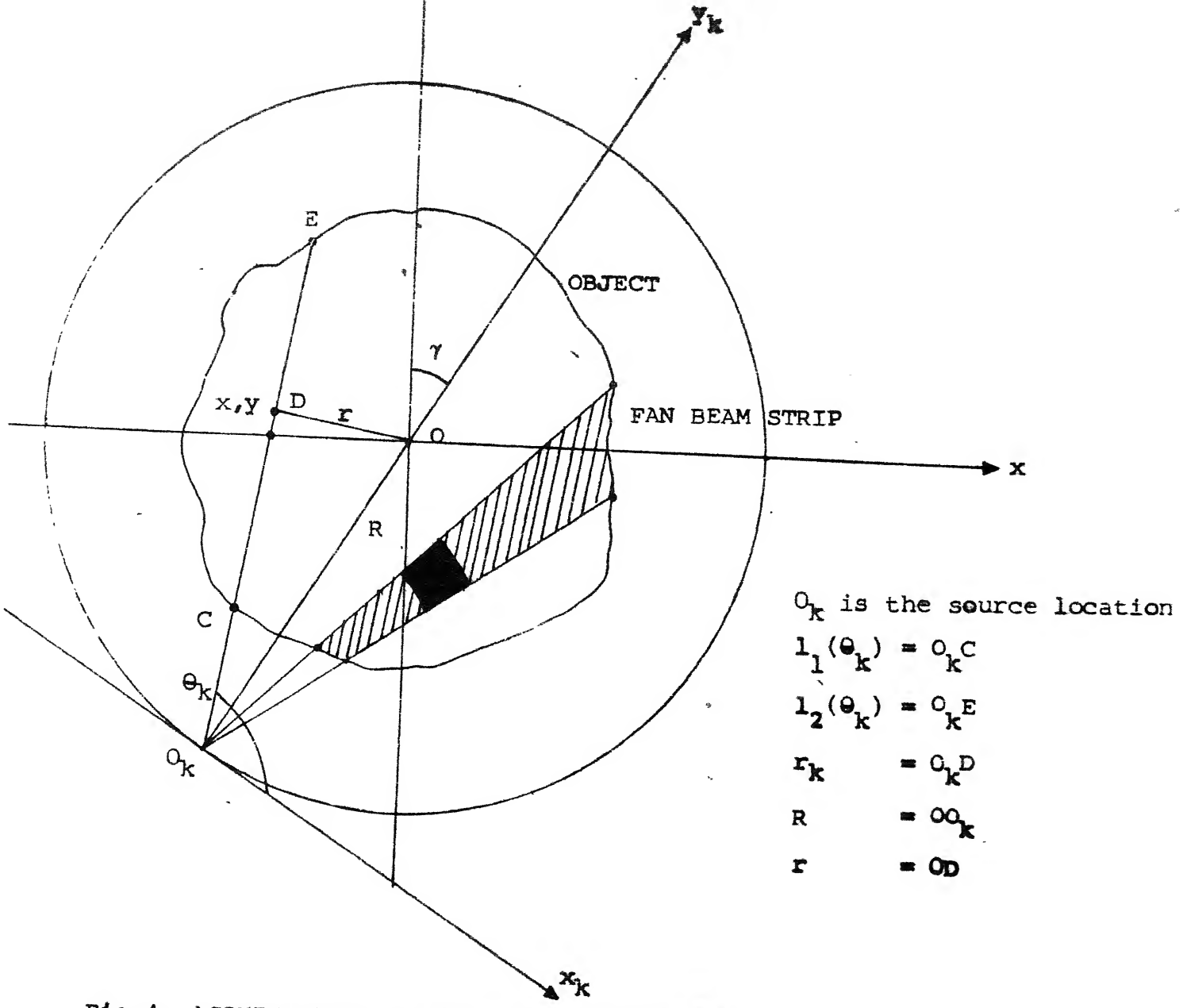
$$\tan\theta_k = \left[\frac{-x\cos\gamma - y\sin\gamma + R}{-x\sin\gamma + y\cos\gamma}\right]$$

and

$$r_k^2 = R^2 + r^2 - 2Rr\cos(\gamma-\theta)$$

R is the radius of the circle on which the source rotates and (r, θ) are the polar coordinates of any point (x,y). θ_k, $l_1(\theta_k)$, $l_2(\theta_k)$ are as shown in the figure (Fig.4). If the projections are taken at discrete angles $[\gamma=\gamma_k,\ k=1,2,..,N]$ then $[B_k(\theta_k)]$ represent the N fan beam projections of the object $[A(x,y)]$.

One of the many points of interests is the extent to which the object $A(x,y)$ can be recovered from its projections. The following theorems have been proved [24] in this connection.



O_k is the source location

$l_1(\theta_k) = O_kC$

$l_2(\theta_k) = O_kE$

$r_k = O_kD$

$R = OO_k$

$r = OD$

Fig.4: ACQUISITION OF FAN BEAM PROJECTIONS.

Theorem 1.1: An object is determined by any infinite set of projections.

Theorem 1.2: A finite set of projections tells nothing at all.

Theorem 1.3: Suppose given an infinitely differentiable object f_o and a finite number of directions. Then there is a new infinitely differentiable object f with exactly the same shape, exactly the same projections from these directions, and completely arbitrary on any compact set in the interior of the support of f_o.

Theorem 1.4: For almost any finite dimensional space F, the objects in F can be distinguished by a single projection from almost any direction.

The second theorem is one of those mathematical ideals untainted by any possibility of practical application. Although a reconstruction method necessarily uses only a finite number of projections, it also necessarily produces only a finite dimensional space of possible reconstructions. The question arises as to whether the projections serve to distinguish between the objects within this finite dimensional space. Theorem 1.4 says that the objects within this space can indeed be distinguished.

So far we have formulated and dealt with the problem in its continuous form. To enable the use of powerful digital processing techniques and the digital computer, the set of equations must be discretized.

For our purpose we have discretized the system so that for the ith projection, we get an equation of the form

$$\sum_{j=1}^{N^2} A_{ij}\, x_j = B_i$$

N^2 is the total number of picture elements (pixels) in a NxN discretized picture.

A_{ij} = 1 or 0 depending on whether the pixel 'x_j' is included in the projection (ray) or not.

B_i is the sum of all pixels included in the particular projection. Thus we obtain a set of linear simultaneous equations

$$\underline{A}\underline{x} = \underline{B}$$

where $\underline{A}$ is a matrix of size MxN^2 and $\underline{B}$ is a vector of size Mx1; M being the total number of projections.

The following points may be noted:

(1) The matrix [A] is quite sparse, since from the geometry of projections, it is obvious that each projection includes very few of the pixels.

(2) The size of the matrix [A] is enormous. In typical applications N^2 starts at 2500 and with little ambition can easily reach 10^6 to 10^9. The number of projection elements ranges from 500 to 10^5 and 10^7 in corresponding cases. Thus the number of elements in the [A] matrix can range between 1.25×10^6 to 10^{16}.

(3) The equations are ordinarily highly underdetermined, i.e. there are less number of equations than there are unknowns.

(4) The rank of matrix [A] is unknown.

(5) The matrix [A] is nonnegative since $A_{ij} \geq 1$ or 0.

(6) The projections B_i are ordinarily nonnegative.

(7) The pixel values x_i are assumed to be nonnegative so that one desires a solution for which $x_i \geq 0$.

(8) The errors in the data may cause the equations to be inconsistent.

(9) Statistical effects of noise in the data should be analysed if possible.

(10) The approximations which lead to the equations $\underline{Ax} = \underline{B}$ introduce systematic errors which have to be analysed. In our case the approximations are:

(a) All activity is assumed to be at the centre of the pixel.

(b) A ray is assumed to be of zero width. No weight has therefore been given either to the area of the pixel intersected by the ray or to the length of the ray that traverses the pixel.

The projection data used for reconstruction is likely to get contaminated by noise. In the case of transmission of pictures, the data is likely to be corrupted by additive noise while it is being transmitted. In biomedical applications the X-rayfilm is affected by noise in the following manner [24].

When a photon from the X-ray beam passes through the object two things can happen. It can sail through completely undisturbed or it can interact with the atoms of the object. In the case of an interaction the photon may be stopped which attenuates the X-ray beam. Unfortunately, however, the interactions are not this simple. Sometimes they involve not the stopping of X-ray photon but a change in its direction. Other times they involve the release of photons from the object atom. These new photons and the old ones with the wrong direction are called scatter. There are many other sources of noise of which a few are: finger marks of the technicians, streaks left by dirty rollers in the developing machine, incorrect developing temperature and uneven film emulsion. The noise may be large and comparable with the difference between the maximum and minimum pixel value. This may necessitate some preprocessing of the raw data.

In all these cases the available data is a set of projections taken at different angles. Thus it is possible to identify a general problem, involving the reconstruction of an image from its projections, which is not restricted to the narrow confines of any specific application.

The aim of this thesis being the application and testing of a certain algorithm for reconstruction, we have not gone into an indepth study of the relative merits of the two types of geometries for obtaining projections, but have restricted ourselves to the gathering of projection data by parallel beam geometry only.

The reconstruction problem has been solved by us by the use of a certain block iterative algorithm. The amount of data to be handled being enormous, data structuring has been done for efficient computation and storage. The algorithm has been verified by taking a digitized picture and obtaining a linear set of simultaneous equations from it. The reconstructed picture has been visually compared with the original picture after printing both of them on a line printer, where the different grey levels have been simulated by overprinting. The mean square error has also been computed.

CHAPTER 2

THE ALGORITHMS USED

In Chapter 1 we discretized the problem to obtain a set of linear simultaneous equations of the form

$$\underline{A}\,\underline{x} = \underline{B}$$

where $\underline{A}$ is a mxn matrix, $\underline{x}$ is a nxl vector and $\underline{B}$ is a mxl vector. The methods for solving this system can be divided into the following categories:

1) Direct matrix inversion techniques; generalized inverse and pseudo inverse [8,12,27,28].
2) Summation, linear superposition or back projection techniques [30].
3) Algebraic reconstruction technique (ART) [5,10,11,15, 16, 17, 30].
4) Simultaneous iterative reconstruction technique (SIRT) [30,32].
5) Minimization of criterion functions [19].
6) Summation of filtered back projection, convolution technique [14, 17, 30].
7) Fourier reconstruction method [9,13,14,17,31].

1) Generalized inverse and pseudo inverse:

A generalized inverse of a mxn real matrix $\underline{A}$ of rank k, is a convenient tool for handling numerically the system $\underline{A}\,\underline{x} = \underline{B}$ of m linear equations in n unknowns. Any

numerical procedure for solving such systems will have to distinguish four cases. Suppose that the rank of $[\underline{A}, \underline{B}]$ is k'

(1) $k = k'$. Then the equations have one or more solutions.

i) If $k = n \leq m$, then there is a unique solution.

ii) If $k < n$, then the equations have an infinite number of solutions that can be written in the form $\underline{x} = \underline{x}_o + \underline{u}$, i.e. $u \in N(\underline{A})$ the null space of $\underline{A}$, where $N(\underline{A})$ is a vector space of dimension $n-k$.

(2) $k < k'$. The equations are inconsistent. In this case it makes sense in many applications to look for a least square solution, i.e. the solution that minimizes $\| \underline{A}\,\underline{x} - \underline{B} \|_2$, the size of the residual $\underline{B} - \underline{A}\,\underline{x}$ in the 2-norm. We again distinguish two cases:

i) If $k = n$, there exists a unique least squares solution.

ii) If $k < n$, the residual is minimized by a family of solutions $\underline{x} = \underline{x}_o + \underline{u}$, where $\underline{x}_o$ is a particular least square solution and $\underline{u} \in N(\underline{A})$.

The generalized inverse is motivated by the desire to cover all of these cases by one solution formula, $\underline{x} = \underline{x}_o + \underline{u}$, $\underline{x}_o = \underline{G}\,\underline{B}$, $\underline{G}$ being the generalized inverse.

Def. 2.1: A g-inverse of an mxn matrix $\underline{A}$ is any matrix $\underline{G}$ such that

$$\underline{A}\,\underline{G}\,\underline{A} = \underline{A}$$

A symmetric g-inverse satisfies $\underline{A}\ \underline{G}\ \underline{A} = \underline{A}$, $(\underline{A}\ \underline{G})^T = \underline{A}\ \underline{G}$.

Def. 2.2: A least squares solution of $\underline{A}\ \underline{x} = \underline{B}$ is a solution that minimizes $\|\underline{A}\ \underline{x} - \underline{B}\|_2$. A minimal least squares solution is one that minimizes $\|\underline{A}\ \underline{x} - \underline{B}\|_2$ and $\|\underline{x}\|_2$.

Theorem 2.1: A necessary and sufficient condition for $\underline{x} = \underline{G}\ \underline{B}$ to be a least squares solution of $\underline{A}\ \underline{x} = \underline{B}$ is that $\underline{G}$ be a symmetric g-inverse.

Theorem 2.2: If $\underline{G}$ is a symmetric g-inverse, necessary and sufficient conditions for $\underline{x} = \underline{G}\ \underline{B}$ to be a minimal least squares solution of $\underline{A}\ \underline{x} = \underline{B}$ is that $\underline{G}$ also satisfy

$$\underline{G}\ \underline{A}\ \underline{G} = \underline{G}\ , \quad (\underline{G}\underline{A})^T = \underline{G}\ \underline{A}\ .$$

Def. 2.3: The Moore-Penrose generalized inverse of $\underline{A}$ is a matrix $\underline{G} = \underline{A}^+$ that satisfies the four conditions

$$\underline{A}\ \underline{G}\ \underline{A} = \underline{A}$$
$$(\underline{A}\ \underline{G})^T = \underline{A}\ \underline{G}$$
$$\underline{G}\ \underline{A}\ \underline{G} = \underline{G}$$
$$(\underline{G}\ \underline{A})^T = \underline{G}\ \underline{A}$$

From the above definitions and theorems, it is clear that the Moore Penrose inverse may be a solution we are looking for. The methods of finding the Moore Penrose inverse are summarized below:

i) If $\underline{A}$ is mxn ($m \neq n$) and it is known to be of full rank, the most straightforward method for computing $\underline{A}^+$ is via

$$\underline{A}^{+} = (\underline{A}^{T}\,\underline{A})^{-1}\,\underline{A}^{T} \quad (m > n) \quad \text{or} \quad \underline{A}^{T}(\underline{A}\,\underline{A}^{T})^{-1} \quad (m < n)$$

This is a direct method. The Greville algorithm is a recursive algorithm for computing the Moore Penrose inverse of a matrix. We present the algorithm without proof.

(ii) Greville Algorithm: Let $\underline{A} = (\underline{a}_1\ \underline{a}_2 \cdots \underline{a}_k \cdots \underline{a}_n)$ where $\underline{a}_k$ denotes the kth column of the given matrix $\underline{A}$ and

$$\underline{A}_k = (\underline{a}_1\ \underline{a}_2 \cdots \underline{a}_k)$$

Consider $\underline{A}_k$ in the partitioned form

$$\underline{A}_k = (\underline{A}_{k-1},\ \underline{a}_k)$$

Compute $\underline{d}_k = \underline{A}_{k-1}^{+}\,\underline{a}_k$

and $\underline{c}_k = \underline{a}_k - \underline{A}_{k-1}\,\underline{d}_k$

If $\underline{c}_k \neq 0$, then $\underline{e}_k = \underline{c}_k^{+} = (\underline{c}_k^{t}\,\underline{c}_k)^{-1}\,\underline{c}_k^{t}$

$$\text{otherwise } \underline{e}_k = (1 + \underline{d}_k^{t}\,\underline{d}_k)^{-1}\,\underline{d}_k^{t}\,\underline{A}_{k-1}^{t}$$

$\underline{A}_k^{+}$ is then given by

$$\underline{A}_k^{+} = \begin{bmatrix} \underline{A}_{k-1}^{+} - \underline{d}_k\,\underline{e}_k \\ \underline{e}_k \end{bmatrix}$$

To start the recursion, take $\underline{A}_1^{+} = 0$ if $\underline{a}_1 = 0$ (null column vector); otherwise,

$$\underline{A}_1^{+} = (\underline{a}_1^{t}\,\underline{a}_1)^{-1}\,\underline{a}_1^{t}$$

This algorithm requires only one decision in each recursion and is matrix inversion free.

2) Linear superposition or back projection

The simplest and most rapid method of reconstructing a two dimensional distribution from multiple one dimensional projections is to merely project the views back to a common object region. Thus we describe back projection as

$$x_i' = \Sigma B_j$$

where the summation takes place over all those rays which have passed through x_i. The total density or concentration for the section or array being reconstructed is estimated by

$$T = \Sigma B_j$$

where this summation is over all rays of a single projection angle. After back projecting, the total density T' for the array is

$$T' = \sum_{1}^{N^2} x_i' , \quad N^2 = n; \text{ the number of pixels.}$$

A normalization factor is derived for reducing the value of each picture element so that the reconstructed array total density corresponds to the estimated total. Thus the corrected back projected image is

$$x_i = x_i'(T/T').$$

A more exact background correction involves modifying the values by subtracting from each pixel the mean density or concentration multiplied by the number of views minus one

$$x_i = x_i' - [T(\text{no.of views} - 1)]/N'$$

where N' is the number of sample points per projection.

3) Algebraic reconstruction technique:

This simple algorithm consists of guessing a value of all the pixels and then compensating for the discrepancy between the measured ray sum B_j and calculated ray sum B'_j,

$$x_i^{n+1} = x_i^n \frac{B_j}{B'_j}$$

If the measured ray sum and the calculated ray sum turn out to be the same then the guessed values are correct for the particular projection. However, for a different projection, there may be a large discrepancy. Thus the pixels ofthe old ray which also lie in the new ray willbe modified again. The above procedure is called multiplicative ART.

There is another form of ART in which the discrepancy between the measured and calculated ray sum is corrected by adding to each pixel the difference between the two divided by the number of pixels in the ray,

$$x_i^{n+1} = \max\left[x_i^n + \frac{B_j - B'_j}{N'_j}, 0\right]$$

N'_j is the number of pixels in the ray which passes through the pixel x_i. This is called additive ART.

There are a few modifications of ART. One consists of setting to zero those values that were zero,since they corrêsponded to a ray sum that was zero. This is a boundary condition for this iterative technique. Another form of ART

takes into account noise and has been used effectively in transmission studies and simulations with noise.

It may be noted that the following unconstrained version of ART is the same as the iterative matrix inversion method by Kacmarz.

$$x_j^{p+1} = x_j^p + [A_{ij}(B_i - \sum_{j=1}^{N^2} A_{ij}\, x_j)]/N_i$$

where $N_i = \sum_{j=1}^{N^2} A_{ij}^2$, $i = 1,2,\ldots m$.

4) Simultaneous iterative reconstruction technique:

The simultaneous iterative reconstruction technique differs from ART in the fact that at each iteration the pixels are modified by using data from all projections simultaneously. Thus

$$x_i'^{n+1} = \max\left[x_i^n + \frac{\Sigma B_j}{\Sigma L_j} - \frac{\Sigma B_j'}{\Sigma N_j}, 0\right]$$

Here the summations are done over those rays which intersect the pixel. L_j is the length of the ray j. B_j is the measured project density of ray j, B_j' is the calculated projected density of ray j, and N_j is the number of pixels in ray j.

A multiplicative algorithm has also been given, which is

$$x_i'^{n+1} = \max\left[\frac{\Sigma B_j\, \Sigma N_j}{\Sigma L_j\, \Sigma B_j'} x_i^n, 0\right]$$

After each iteration the total array is normalised such that for all i $x_i^{n+1} = (T/T')\, x_i'^{n+1}$, where T and T' have been defined before.

5) Minimization of criterion functions:

An example of this method is the algorithm suggested by Kashyap and Mittal in which sum of two quadratic functions J_1 and J_2 has been minimized subject to the condition $\underline{A}\,\underline{x} = \underline{B}$. J_1 is the nonuniformity function

$$\frac{1}{2} \sum_i (x_i - \bar{x}_i)^2 = \frac{1}{2} \underline{x}^T \underline{c}\, \underline{x} ,$$

i consists of all cells not on the boundary, $\bar{x}_i$ is the average of eight neighbours of cell i and $\underline{c}$ is a positive definite matrix. J_2 is the loss function

$$\frac{1}{2} \sum_{i=1}^{N^2} (x_i - a)^2$$

where a is the mean of intensities of the N^2 cells.

$$J_1 + J_2 = \frac{1}{2} \underline{x}^T \underline{c}\, \underline{x} + \frac{1}{2} \underline{x}^T \underline{x} - \frac{1}{2} N^2 a^2$$

$$= J_3 - \frac{1}{2} N^2 a^2$$

If we assume that a is known, then the problem is to minimize J_3 subject to $\underline{A}\,\underline{x} = \underline{B}$. For this a function $J(\underline{x})$ is chosen

$$J(\underline{x}) = \frac{1}{2}\gamma \underline{x}^T \underline{c}\, \underline{x} + \frac{1}{2} \underline{x}^T + \frac{1}{2}\beta \| \underline{B} - \underline{A}\, \underline{x} \|^2$$

in which γ can be chosen experimentally and β is a large positive integer. $J(\underline{x})$ is minimized by a recursive method, the steepest descent method.

$$\underline{x}(t+1) = \underline{x}(t) - (p \delta J / \delta \underline{x}) \Big|_{\underline{x} = \underline{x}(t)}$$

$$= \underline{x}(t) - p[\underline{x}(t) + \gamma\, \underline{c}\, \underline{x}(t) - \beta\, \underline{B}^T (\underline{y} - \underline{B}\, \underline{x}(t))]$$

$$t = 0,1,2\ldots$$

p is a scalar gain which can be chosen to ensure proper rate of convergence.

6) Fourier Reconstruction Method

To describe the Fourier reconstruction method, we shall introduce a new set of notations. Let $f(x,y)$ or $f(r,\emptyset)$ be the intensity distribution of the object and $g(l, \theta)$ be the projection at position l at angle θ. Then $f(x,y)$ or $f(r,\emptyset)$ can be expressed in terms of $g(l,\theta)$ by the following Fourier transform relationships.

$$F(R, \theta) = \int_{-\infty}^{\infty} g(l,\theta) \exp(-2\pi i R l) dl \qquad (2.1)$$

$$f(r,\emptyset) = \int_{0}^{2\pi} \int_{0}^{\infty} F(R,\theta) \exp[2\pi i R r \cos(\emptyset-\theta)] R dR d\theta \qquad (2.2)$$

The Fourier reconstruction method consists of either directly evaluating these integrals or using one of the methods described below:

The basic equations can be rewritten in a form not involving Fourier transforms, but containing only integrals of functions defined in the real space of observation. We can write eqn.(2.2) as

$$f(r,\emptyset) = \int_{0}^{\pi} \int_{-\infty}^{\infty} |R| F(R,\theta) \exp[2\pi i R r \cos(\emptyset-\theta)] dR d\theta \qquad (2.3)$$

If we define,

$$g'(l,\theta) = \int_{-\infty}^{\infty} |R| F(R,\theta) \exp(2\pi i R l) dR \qquad (2.4)$$

then from equation (2.3),

$$f(r, \emptyset) = \int_{0}^{\pi} g'[r\cos(\emptyset-\theta),\theta] \, d\theta \qquad (2.5)$$

Fourier inverting equation (2.1) we get

$$g(l,\theta) = \int_{-\infty}^{\infty} F(R, \theta) \exp[2\pi i R l] \, dR \qquad (2.6)$$

Comparing equations (2.4)and (2.5) we obtain

$$\text{F.T. of } g'(l,\theta) = \text{F.T. of } g(l,\theta) \times \text{F.T. of } q(l) \qquad (2.7)$$

where $|R|$ is the Fourier transform of q(l) or

$$|R| = \int_{-\infty}^{\infty} q(l) \exp(2\pi i R l) dl$$

It may be noted that back projecting $g'(l,\theta)$ will give us the true image (eqn.2.6). There are two methods to evaluate $g'(l,\theta)$.

a) Convolution Method:

From the convolution theorem for the inverse of the product of Fourier transforms and equation (2.7),we can write

$$g'(l,\theta) = \int_{-\infty}^{\infty} g(l_1,\theta) \, q(l-l_1) \, dl_1 \qquad (2.8)$$

For calculating $g'(l,\theta)$ we require q(l) explicitly. Now,

$$q(l) = \int_{-\infty}^{\infty} |R| \exp(-2\pi \, i \, R \, l) \, dR$$

This integral cannot be evaluated between the limits $-\infty$ to ∞ as the integrand diverges. So to overcome this difficulty, we replace $-\infty$ and $+\infty$ by $-A/2$ and $+A/2$ where A is a finite number.

$$q_A(l) = \int_{-A/2}^{A/2} |R| \exp(2\pi \, i \, R \, l) dR$$

Thus we get, $q_A(na) = 1/4a^2$ for $n = 0$

$= -1/\pi^2 n^2 a^2$ for n odd

$= 0$ for n even.

q(na) being a set of points at equal spacing a.

Equation (2.8) can be expressed in the form of an infinite sum

$$g'(na,\theta) = a \sum_{m=-\infty}^{\infty} g(ma, \theta)\, q[(m-n)a]$$

$$= g(na, \theta)/4a^2 - (1/\pi^2) \sum_{p \text{ odd}} g[(n+p)a,\theta]/p^2$$

The true image can now be generated by back projecting $g'(na, \theta)$.

Filtered back projection:

The second method, called filtered back projection, starts with the following equation,

$$\text{F.T. of } g'(1,\theta) = \text{F.T. of } g(1, \theta) \times |R|$$

Sometimes instead of $|R|$ we have $w(R).|R|$, where w(R) is a window function. The R.H.S. is now inverse Fourier transformed to give $g'(1,\theta)$. This is now back projected to yield the true image.

It may be noted that both these methods are essentially the same. The first method carries out the operations in the real space of observation whereas the second method operates in the frequency space.

The Method Used

This algorithm proposed by Rathore, is a general stationary least square method to solve a rectangular set of equations. This is an iterative method and its convergence under very general conditions has been established. Even for inconsistent systems, the method provides a least square solution. One good feature of this method is its guaranteed convergence. The method is now described.

Consider the equation $\underline{A}\ \underline{x} = \underline{B}$. An asumed vector $\underline{x}^{o}$ will give rise to a residual error vector $\underline{r}_{o}^{o}$ given by

$$\underline{r}_{o}^{o} = \underline{B} - \underline{A}\ \underline{x}^{o}$$

Thus $$\underline{r}_{o}^{o} = \underline{A}[\underline{x} - \underline{x}^{o}]$$

$$= \underline{A}\ \underline{e}_{o}^{o}$$

where $\underline{e}_{o}^{o}$ is the error vector.

If we can somehow calculate the error vector approximately, we get a improved solution

$$\underline{x}^{1} = \underline{x}^{o} + \underline{e}_{1}^{o}$$

where $\underline{e}_{1}^{o}$ is an approximate solution of $\underline{r}_{o}^{o} = \underline{A}\ \underline{e}_{o}^{o}$.

The iterative method may be continued and the process has been shown to converge. The method of residual projections computes the error vector by projecting the residual at every stage on the columns of matrix $\underline{A}$. A heuristic reason for doing so is as follows:

The solution $\underline{x}$ of the problem satisfies

$$\underline{B} = \underline{A}\,\underline{x}$$

$$\underline{B} = \sum_{j=1}^{n} [\underline{a}_j]\, x_j$$

where $\underline{a}_j$ are the column vectors of $\underline{A}$. The values x_i may be assumed to be the contributions of each column vector towards the making of $\underline{B}$. The solution $\underline{x}$ is a set of contributions for which the residual $\underline{B}-\underline{A}\,\underline{x}$ is minimized in a least square sense. When the residual is orthogonalized with respect to the first column vector of $\underline{A}$, it means that the component of the first column vector along the residual is zero. This is done for every column of $\underline{A}$. The vector $\underline{x}$ for which the residual has no component along any of the columns of $\underline{A}$ is the required solution.

The residual projection algorithm consists of the following steps:

a) Start with any arbitrary initial vector and compute the corresponding residual.

b) Project the residual computed in (a) on the first column of matrix $\underline{A}$ and compute its component along first column.

c) Alter the first entry in the initial vector by adding the scalar component of the residual computed in (b) to it.

d) Remove the vector component of the residual along the first column to obtain new residual.

e) Project the residual obtained in (d) on the second column

of $\underline{A}$ and update the second entry in the solution vector by adding the scalar component of the residual along the second column to it. Subtract the vector component of the residual along the second column from the residual obtained in (b).

f) Continue the process till the last column is finished.

g) Go to (b) and repeat the process through (f) till satisfied.

These steps can be mathematically expressed as follows:

i) Start with an initial vector $\underline{x}^{o}$

ii) Compute the residual $\underline{r}_{o}^{o} = \underline{B} - \underline{A}\,\underline{x}^{o}$

Then for each k = 0,1,2,... till satisfied do,

iii) For each j = 1,2,...N do

$$\underline{r}_{j}^{k} = \underline{r}_{j-1}^{k} - [(\underline{r}_{j-1}^{k}, \underline{a}_{j})/(\underline{a}_{j}, \underline{a}_{j})]\,\underline{a}_{j}$$

and $$x_{j}^{k+1} = x_{j}^{k} + [(\underline{r}_{j-1}^{k}, \underline{a}_{j})/(\underline{a}_{j}, \underline{a}_{j})]$$

k keeps track of the number of iterations and j keeps track of the column on which the residual is being projected.

The following theorem has been proved.

Theorem 2.4: The vector sequences $[\underline{r}^{(i)}]$ i = 0...∞ and $[\underline{x}^{(i)}]$ i = 0 ... ∞, generated by the method of residual projections converge for arbitrary $\underline{A}$, $\underline{B}$ and $\underline{x}(o)$ with $\underline{r}^{(\infty)} = \lim_{i \to \infty} \underline{r}^{(i)} = \underline{P}_{k}\,\underline{B}$ and $\underline{x}^{(\infty)} = \lim_{i \to \infty} \underline{x}^{(i)} = \underline{P}\underline{x}^{(o)} + \underline{G}\,\underline{B}$.

Moreover, $\underline{x}^{(\infty)}$ minimizes $\underline{B}-\underline{A}\,\underline{x}$ and $\underline{G}$ is a reflesive least squares g inverse of $\underline{A}$. $\underline{P}$ is the projection onto kernel of $\underline{A}$ along image of $\underline{G}\,\underline{A}$ and $\underline{P}_k$ is the orthogonal projection onto kernel of $\underline{A}$.

Residual projection with averaging

Since the system of equations taken for reconstruction was almost always underdetermined, the least square solution provided by the residual projection algorithm did not always coincide with the desired solution. The restoration often resulted in 'ghosts'; pictures entirely different from the actual picture. To resolve this problem, additional information had to be provided in the form of physical constraints.

Inspired by the idea of Kashyap and Mittal [19] the idea of continuity was applied to our method. Any real picture is continuous and the value of each pixel is nearly equal to the average of the pixels surrounding it. But this property, as applied by Kashyap and Mittal, gave rise to large matrices whose elements were other than one and zero. Thus efficient computation and storage was impossible. Each iteration took 30 to 40 seconds by their method. Their idea was modified to suit our purpose.

Instead of minimizing the quantity

$$J = \frac{1}{2} \sum_{i \in s} (x_i - \bar{x}_i)^2$$

where s is the set of pixels not on the boundary and $\bar{x}_i$ is the average of the neighbours around x_i, we tried to force each pixel (not on the boundary) to a value which was the average of all the eight neighbours around it. This was done after each iteration and the values of $\underline{x}$ thus computed were used to find the new residual $\underline{B}-\underline{A}\,\underline{x}$. It may be noted that the implementation of this scheme is extremely simple, needs almost no storage and is extremely fast.

Thus after each iteration we have

$$x_i = \frac{1}{8} \Sigma \text{ Neighbours of } x_i,$$

where x_i is not a boundary pixel. Thus we get the following matrix equation

$$\underline{x} = \underline{C}\,\underline{x}$$

where $\underline{C}$ is a $N^2 \times N^2$ matrix and N^2 is the total number of pixels. The elements of $\underline{C}$ are as follows.

If x_i is a boundary cell then

$$C_{ij} = 0 \quad \text{for } i \neq j$$
$$= 1 \quad \text{for } i = j$$

If x_i is not a boundary cell then

$$C_{ij} = 1/8 \quad \text{for} \quad j = i-N-1;\ i-N, i-N+1, i-1, i+1, i+N-1, i+N, i+N+1$$
$$= 0 \quad \text{for all other values of } j.$$

This operation has been used in conjunction with the residual projection method. The overall method converges

if the iteration matrix of the compound operation has a spectral radius less than 1.

Residual Projection with Median Filtering

A variation of the strategy mentioned earlier is to do median filtering at each iteration. In recent years median filters have come into use for noise reduction purposes in signal and picture processing. The main merit of a moving median is that it preserves edges, in constrast to a moving average which smooths edges.

The median of n numbers $x_1 \ldots x_n$ is, for n odd, the middle number in size and we denote it by Median $(x_1 \ldots x_n)$. For example the median of 0, 3, 4, 2, 7 is 3.

A two dimensional median filter with filter window A on a picture$[x_{ij}, (i,j) \in z^2]$ is defined through

$$y_{ij} = \underset{A}{\text{Median}}\, x_{ij} = \text{Median}\, (x_{i+r,j+s};\ (r,s) \in A),\ (i,j) \in z^2$$

Various forms of filter windows A can be used, e.g. line segments, squares, circles and crosses. For such symmetric filter windows, the median filter is expected to preserve edges. The median filter also removes impulse and salt and pepper noise if the rate of error in the noisy picture is low.

In our experiments, a 3x3 rectangular window was used. Since median filtering preserves edges, so should iterations of median filtering. Median filtering was done at each

iteration of the residual projection method to arrive at the exhibited result.

Computational Efforts

The computational efforts required to carry out the different methods of reconstruction have not yet been calculated in terms of the picture size, the number of equations and similar parameters. An effort has been made by us in this direction.

Let the picture size be NxN. Let m be the number of angles and k be the number of rays per angle. k is usually taken to be equal to N. The total number of equations is mk. Since the matrix $\underline{A}$ is sparse, it is essential that the computations be carried out with the nonzero entries only. A software implementation of such a strategy is presented in the next chapter.

An important problem is to first relate the number of nonzero entries 'p' of the $\underline{A}$ matrix to the parameters N, k and m. When the rays are parallel to the sides of the picture we expect that each pixel will be intersected by one ray in a particular angle. For such angles the number of non-zero entries per angle is N^2.

For other angles some pixels will be left out. If the ratio of the area of the picture not intersected by any ray to the total area of the picture is α, then the number

of pixels not intersected by any ray will roughly be αN^2. Simple trigonometric calculations will show that this ratio α_β for any angle β is

$$\alpha_\beta = \frac{\tan \beta}{2} \left[1 - \tan \beta + \frac{1 - \cos \beta}{\cos\beta \sin\beta}\right]^2$$

The value of α is maximum when the angle β is 45°, and this value is 0.086. Therefore p will be of the order N^2m. This has been verified in our experiments. With a 32 x 32 picture and 17 angles we had $16x10^3$ nonzero entries. Similarly with a 16x16 picture and 9 angles the number of nonzero elements were $3.5x10^3$.

Having given a rough estimate of p we show the computational effort of each method in terms of p.

1) ART: This method requires p multiplications and p additions for calculating the ray sums and p multiplications for modifying the pixel values. This is for a single iteration.

2) SIRT: The computational effort for SIRT is difficult to calculate. We give the figures for the worst case when all the rows of the A matrix contain either all zeroes or ones. Then pN^2 multiplications and pN^2 additions are required to calculate the sum of ray sums and N^2 multiplications are needed to modify the pixel values. This is also for a single iteration.

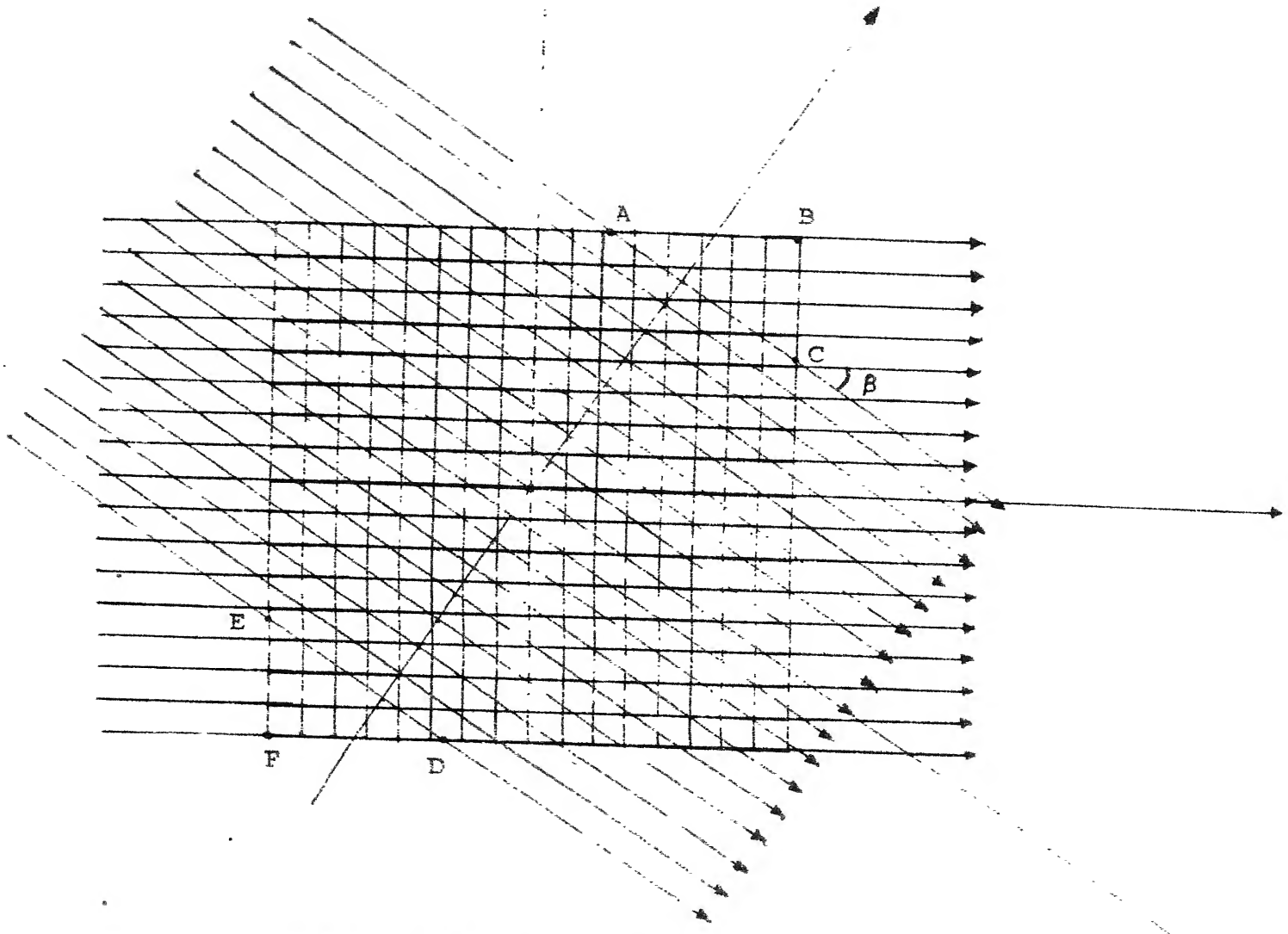


Fig.1: ABC and DEF are the two regions of the picture, which are not intersected by any ray at angle β.

3) Back Projection: This method requires p multiplications and p additions for projection. In addition p multiplications and p additions more are required to calculate the normalization factor. Thus 4p operations are required in total.

4) Convolution Method: Here the computational requirement is k^2m multiplications and k^2m additions for convolution and 4p operations for back projection.

5) Filtered Back Projection: The filtered back projection method requires $\frac{km}{2} \log_2 km$ complex multiplications and $km \log_2 km$ complex additions for computing the FFT of projections, km multiplications for filtering, $\frac{km}{2}\log_2 km$ complex multiplications and $km \log_2 km$ complex additions for computing IFFT and 4p operations for back projection.

6) Residual Projection Method: This method requires p multiplications and p additions for projecting the residual on the columns of A matrix, p additions for modifying the residuals and N^2 multiplications for modifying the pixel values. These are the values for a single iteration.

Reconstruction with noise: Since the process of averaging reduces noise, the residual projection method along with averaging should be ableto reconstruct pictures in the presence of noise.

We expreimented by adding uniform random noise between 0 and 1 to a picture whose maximum intensity was 5. Projections were obtained from the resulting picture and an attempt was made to reconstruct the original picture from these projections.

CHAPTER 3

SOFTWARE PROBLEMS

The major hurdles in a reconstruction problem are computation and storage. Even considering a 64 x 64 picture, the number of unknowns is 4096 and the weighting matrix A is 4096 x 4096, if the system is to be of exact determinacy. Direct inversion techniques are therefore out of question. The method must use the total available data in parts. Also, the sparsity of the A matrix has to be made use of.

As an example consider the case of a 16 x 16 picture solved by the method of residual projections. The number of angles were 9 and the number of rays per angle were 16, which gave a total of 144 equations. Therefore a A matrix whose size 256 x 144 i.e.36864 elements had to be stored. It can easily be seen that increasing the picture size further can pose enormous problems.

A way of resolving this problem is to either produce or read each column separately into the core, use it for computation and then call the next column. The residual projection method is particularly suited for such a scheme. But then this means that there are 256 read statement executions or subroutine calls for each iteration. Since a disk read or a subroutine call takes much more time than a basic operation like multiplication, the whole process takes

enormous time. Thus we have saved in memory space at the expense of time.

If we can load only the nonzero entries of the A matrix into the core and devise a method in which the only operations are performed with these entries, then a saving in time and memory space can be achieved. How this is exactly done is shown below.

Advantage is taken of the binary nature of the A matrix. The elements of the A matrix are numbered thus:

$$\begin{bmatrix} 1 & M+1 & \cdots & (N^2-1)M+1 \\ 2 & M+2 & & \vdots \\ 3 & \vdots & & \vdots \\ 4 & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ M & 2M & & N^2M \end{bmatrix}$$

Now the A matrix is produced columnwise and if, say, the 5th element is the first nonzero element (1) then the first entry of a one dimensional matrix IA i.e. IA(1) = 5. If the second nonzero entry is the M+1th element then IA(2) = M+1. Therefore if a A matrix is of the form

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Then it will be stored in the IA vector in the form IA(1,2,3,4,5) = 1,5,7,8,13. Storing in this manner, we have found that the total size of the IA vector in the case of a 16 x 16 picture has been approximately equal to 3550. A reduction of 1/10 has been achieved. A slight modification has to be done to the IA vector to find out whether the last entry of the A matrix has been reached. One more entry whose value is more than N^2M, is added to the vector IA. In the previous example, the IA vector will be IA(1,2,3,4,5,6) = 1,5,7,8,13,21.

To show how such a storage mechanism could be used in our algorithm, we have shown the steps for one iteration in details.

The steps, expressed mathematically, are

For $j = 1$ to N^2 do

1) $\delta_j = \underline{D}_j^T \underline{r}_{j-1}$

where $\underline{D}_j$ is a column vector of matrix $\underline{D}$ and is equal to $p_j \underline{a}_j$. p_j is a scalar.

2) $x_j = x_j - \delta_j$

3) $\underline{r}_j = \underline{r}_{j-1} - \delta_j \underline{a}_j$

In algorithmic language and using our storage strategy we write the same thing as follows:

1) $m' = 1$

2) Do for $j = 1$ to N^2 the steps 2 through 17.

3) $n' = m'$

4) $\delta_j = 0$

5) $j' = j \times M$

6) Compare IA(m') with j'

if $IA(m') \leq j'$ go to step 7

if $IA(m') > j'$ go to step 11

7) $i' = IA(m') - (j-1)\ M$

8) $\delta_j = \delta_j + r_{i'}\ p_j$

9) $m' = m' + 1$

10) Go to step 6

11) $x_j = x_j + \delta_j$

12) Compare IA(n') with j'

if $IA(n') \leq j'$ go to step 13

if $IA(n') > j'$ go to step 17

13) $i' = IA(n') - (j-1)\ M$

14) $r_{i'} = r_{i'} - \delta_j$

16) $n' = n' + 1$

17) Continue

Similar strategies have been adopted for carrying out all the operations prior to starting the iterations. It should be noted that all matrix operations like matrix multiplication etc. must essentially be done columnwise. This strategy reduces the number of computations from

$O(N^2M)$ to $O(kN^2M)$ where k is a fraction. (In our example, k = 0.1 for the 16 x16 picture and 0.01 for the 64 x 64 one). The value of k depends on the picture and the angles of projection. Further reduction may be made if a symmetry is noted in the position of the entries of the A matrix. For the pictures and the particular projection angles that we have taken no such symmetry was observed.

It may be noted that even if attenuation correction schemes are used, resulting in the entries of the weighting matrix [A] being other than 1 or 0, the order of computations or storage will not change. An additional matrix VALIA whose size is the same as IA will have to be used to store the values of the different nonzero entries of the A matrix. The matrix IA ill still indicate the position of nonzero entries only.

CHAPTER 4

ENHANCEMENT

When we solve an underdetermined system using the iterative algorithms, the reconstruction in some cases may turn out to be poor. Furthermore, when a physical constraint like continuity is applied, the picture may be blurred and the edges may loose their sharpness. This is all the more true in our case since each pixel is the average of the points around it. So for visual satisfaction some operations must be done on the reconstructed picture.

Several methods have been tried by us to varying degrees of success. Visual examination revealed the following errors in our reconstructed picture:

1) The edges were not sharp due to the process of averaging.
2) The rest of the picture was riddled withsalt and pepper noise (isolated dark points in light regions and vice-versa).

Thus the enhancement method had to do the following:

1) Smooth the picture without blurring the edges.
2) Sharpen the edges.

Removal of salt and pepper noise[3]: To smooth a picture without blurring it, one can use nonlinear operations that involve averaging. The simplest class of such operations combines averaging with thresholding. One can clean up salt and pepper noise by making a point 'whiter' if the average

gray level in some neighbourhood of it exceeds its gray level by more than some threshold. The procedure is similar for making a point blacker.

We have tried a simple implementation of this scheme. For a particular pixel x_j, let the average of its eight neighbours be $\bar{x}_j$. Now, if the quantity $x_j - \bar{x}_j$ is higher than 1/4th of the maximum possible value, then x_j is replaced by $\bar{x}_j$; otherwise x_j is left unchanged. This process does not blur the picture but tends to smooth it out.

Sharpening the edge[3]: Since integration (or averaging) blurs a picture, a natural approach to deblurring is to perform some sort of differentiation operation. If deblurring is required only in some particular direction, one can simply take a directional derivative. For most purposes, however, one wants to deblur in every direction; this can be accomplished by taking the derivative in the gradient direction, i.e. the direction in which the grey levels change fastest at each point. For a well behaved function, this maximal directional derivative is equal to the square root of the sum of squares of the derivatives in any pair of orthogonal directions, e.g. $[(\delta f/\delta x)^2 + (\delta f/\delta y)^2]^{1/2}$. For a digital picture one would approximate the derivatives by differences, e.g.

$$[(a_{i+1,j} - a_{ij})^2 + (a_{i,j+1} - a_{ij})^2]^{1/2}$$

Similar results are obtained by using a sum of absolute values of differences rather than the square root of the sum of their squares. Typical digital gradients are

a) $$\Big([(a_{i-1,j-1} + a_{i-1,j} + a_{i-1,j+1}) - (a_{i+1,j-1} + a_{i+1,j} + a_{i+1,j+1})$$
$$+ [(a_{i-1,j-1} + a_{i,j-1} + a_{i+1,j-1})$$
$$- (a_{i-1,j+1} + a_{i,j+1} + a_{i+1,j+1})]^2\Big)^{1/2}$$

b) $$(a_{i-1,j-1} + a_{i-1,j} + a_{i-1,j+1}) - (a_{i+1,j-1} + a_{i+1,j} + a_{i+1,j+1})$$
$$+ (a_{i-1,j-1} + a_{i,j-1} + a_{i+1,j-1})$$
$$- (a_{i-1,j+1} + a_{i,j+1} + a_{i+1,j+1})$$

Another useful combination of derivatives is the Laplacian $(\delta^2 f/\delta x^2) + (\delta^2 f/\delta y^2)$. For a digital picture the analogous expression is

$$[(a_{i,j} - a_{i-1,j}) - (a_{i+1,j} - a_{ij})]$$
$$+ [(a_{ij} - a_{i,j-1}) - (a_{i,j+1} - a_{ij})]$$
$$= 4a_{ij} - (a_{i-1,j} + a_{i+1,j} + a_{i,j-1} + a_{i,j+1})$$

Using the Laplacian a filter was made, which was supposed to enhance the edges. The filter performed the following operation:

$$y(x_1, x_2) = x(x_1, x_2) + 0.2\left(\frac{\delta^2}{\delta x_1^2} + \frac{\delta^2}{\delta x_2^2}\right)x(x_1, x_2).$$

Removal of negative entries: Since we worked with an underdetermined system, the solution was not exact and some of the pixels which had lower values in the original picture, even turned negative. These pixels usually formed the background and to clean it up we replaced all negative entries by zeroes. A variation of this is to replace negative entries by zeroes at each iteration.

Edge preserving smoothing by using masks[2]: There have been many papers on the subject of smoothing a digital image. A basic trouble with smoothing is that it tends to blur any sharp edges which happen to be present. This edge preserving smoothing method attempts to resolve the conflict between noise elimination and edge degradation. It looks for the most homogeneous region around each point in a picture and then gives each point the average gray level of the selected neighborhood area. Noise in a picture is removed by repeated use of this method, while the edges remain sharp.

The procedure of the edge preserving smoothing is as follows:

1) Rotate an elongated bar mask around a point (x,y).
2) Detect the position of the mask whose variance of the gray level is minimum.
3) Give the average gray level of the mask at the selected position to the point (x,y).
4) Apply steps 1-3 to all points in the picture.

5) Iterate the above method till the gray levels of the picture do not change.

In order to remove the noise without blurring a sharp edge, averaging must not be applied to the area which contains the edge. If an area contains a sharp edge, the variance of the gray level in that area becomes large. Therefore we can use the variance as the measure of nonhomogeneity of an area.

Suppose that a picture has two regions R_1 and R_2, whose mean and variances are $(0, \sigma_1^2)$ and (m, σ_2^2) respectively. Let a point (x,y) belong to R_1. If (x,y) is located in the central portion of R_1, its gray level approaches the gray level of R_1 after several iteration. However, if (x,y) is near the boundary of R_1 there exists two kinds of neighbourhoods, one which completely includes R_1 and the other which contains part of R_1 and R_2. The variance of the former is about σ_1^2. The variance of the latter can be calculated as follows. Let N_1 and N_2 denote the number of points in R_1 and R_2 respectively, which are contained in this neighbourhood. Then the variance

$$\sigma^2 \quad \frac{1}{N}\left[\sum_{i=1}^{N_1}\left(x_i^1 - \frac{N_2}{N}m\right)^2 + \sum_{j=1}^{N_2}\left(x_j^2 - \frac{N_2}{N}m\right)^2\right]$$

where $N = N_1 + N_2$ and x_i^1 and x_j^2 denote the gray levels of the points belonging to R_1 and R_2 respectively. Expanding the R.H.S., we have

$$\sigma^2 = \frac{1}{N}\left[N_1\sigma_1^2 + N_2\sigma_2^2 + \frac{N_1N_2}{N}m^2\right]$$

where we used $\sum_{i=1}^{N_1} {}^{1}(x_i)^2 = N_1\sigma_1^2$,

$$\sum_{i=1}^{N} {}^{1}x_i = 0,$$

$$\sum_{j=1}^{N_2} ({}^{2}x_j - m)^2 = N_2\,\sigma_2^2,$$

and $$\sum_{j=1}^{N_2} ({}^{2}x_j) = N_2 m.$$

If $\sigma^2 < \sigma_1^2$, that is $\sigma_2^2 + (N_1/N)m^2 < \sigma_1^2$ the neighbourhood containing both parts of R_1 and R_2 are selected. Then the boundary between R_1 and R_2 is blurred by this operation. But in most pictures, it is reasonable to assume $\sigma_1^2 = \sigma_2^2$ and $\sigma^2 > \sigma_1^2$. Then the correct neighbourhood is selected.

For actual realization, we have used the nine masks shown in the figure (Fig. 1). The variances of these nine masks are compared with each other and the average gray level of the least variance mask is given to the point (x,y).

The fact that the above method not only smoothens but also sharpens a blurred edge can be understood by considering the following one dimensional example. The following is a one dimensional digital blurred edge

Gray level	0	0	0	0	1	3	5	8	8	8
Mean	0	0	0	1/3	4/3	3	16/3	7	8	8
Variance (x 3)	0	0	0	4/9	14/3	8	38/3	6	0	0

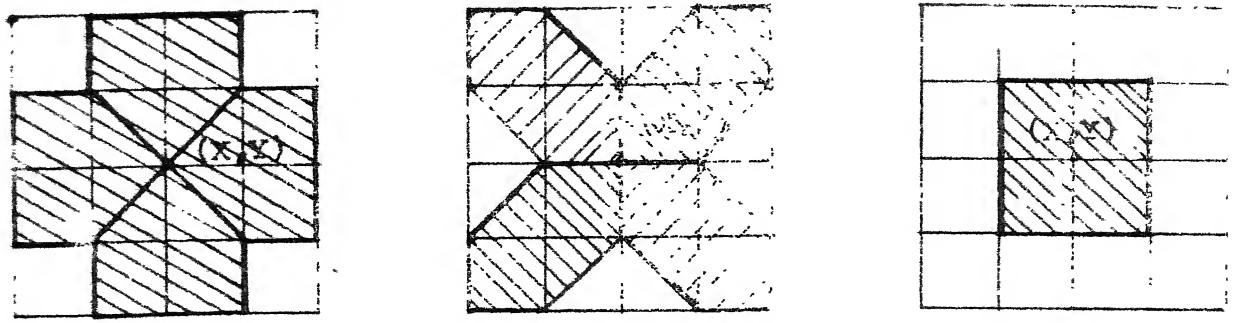
The mean and variance have been calculated using the 1x3 mask centered at each point. If we select the minimum variance mask and give the average gray level of the selected mask to each point we get the following:

0 0 0 0 0 1 7 8 8 8

where the average gray levels have been rounded off to the nearest integer. This, repeated once again, gives a completely sharpened edge.

0 0 0 0 0 0 8 8 8 8

The fluctuation of the gray levels is gradually reduced by several iterations of smoothing. Once a point has a neighbourhood of a constant gray level, its gray level is never changed by smoothing. Therefore, the number of points whose gray levels are changed by smoothing will gradually decrease to zero. The number of iterations needed for convergence depends on the amplitude of noise fluctuation and the shape of regions in the picture.



Discrete realization of the bar masks; 4 pentagonal and 4 hexagonal masks have sharp corners at the centre point (X,Y). The 3 x 3 rectangular mask is used to smooth a small region.

Fig.1

CHAPTER 5

THE RESULTS

The results of the various simulations carried out are presented in this chapter. The programs were run on the DEC 1090 computer and the pictures printed on the line printer. The gray levels were simulated by overprinting different sets of characters. A mean square error criterion was also computed for each of the reconstructed pictures. If x_{ij} and y_{ij} are the original and reconstructed pictures of size NxN, then the mean square error MNSQR is given by

$$\text{MNSQR} = \frac{1}{N^2} \sqrt{\left[\sum_{i=1}^{N} \sum_{j=1}^{N} (x_{ij} - y_{ij})^2 \right]}$$

The error criterion MNSQR gives a reasonable estimate of the accuracy of reconstruction.

The residual projection algorithm guarantees a decrease of the residual $\| \underline{B} - \underline{A}\underline{x} \|_2$. This has been verified and the residual has been tabulated in Table 1. This is for the 32 x32 picture of Fig.1.

Various initial conditions were tried. The one that we found most satisfactory was the following. For each pixel the number of ones in the corresponding column of the A matrix is counted. This number multiplied by 0.1 is the starting point for the particular pixel.

TABLE 1

Iteration No.	Residual	Iteration No.	Residual
1	546.01	16	15.85
2	352.14	17	12.56
3	256.14	18	10.94
4	196.47	19	8.92
5	166.92	20	7.41
6	128.82	21	6.68
7	107.83	22	5.98
8	85.27	23	5.57
9	69.16	24	5.20
10	59.10	25	4.83
11	45.72	26	4.60
12	38.15	27	4.41
13	31.16	28	4.27
14	23.33	29	4.17
15	19.65	30	4.10

Figs. 1 to 5 are the original pictures used for reconstruction. Fig.2 is of size 16x16 and the rest are 32 x 32 pictures. Fig.6 shows the result of reconstructing Fig.1 without averaging. The reconstruction is poor in quality and the MNSQR error is 0.0587 which is comparatively large. Figs. 7 to 11 show reconstructions with averaging. The results are tabulated in Table 2.

TABLE 2

Serial No.	Original picture	Reconstructed picture	Size	MNSQR
1	Fig.1	Fig.7	32x32	0.0441
2	Fig.2	Fig.8	16x16	0.0519
3	Fig.3	Fig.9	32x32	0.0461
4	Fig.4	Fig.10	32x32	0.0408
5	Fig.5	Fig.11	32x32	0.0404

Table 3 shows the results of increasing the number of equations by increasing the number of angles. The original picture is Fig.1.

TABLE 3

S.No.	No. of angles	Rays per angle	No. of equations	Reconstructed pic.	MNSQR
1	13	32	416	Fig.12	0.0322
2	15	32	480	Fig.13	0.0383
3	17	32	544	Fig.7	0.0441
4	19	32	608	Fig.14	0.0465
5	21	32	672	Fig;15	0.0510

The actual C.P.U. time required for one iteration of our method is 0.51 seconds which compares favourably with other methods.

The averages considered so far have been 8 point averages. 4 point averages (Fig. 5.1 and 5.2) were also considered. The reconstructions are shown in Figs. 16 and 17. The MNSQR errors are 0.0457 and 0.0394.

Fig. 18 is the result of doing median filtering at each iteration of residual projection. The result is good, especially from the edge preservation point of view.

Figs. 19 -22 show the results of the various enhancement schemes. These enhancement schemes were carried out for the reconstruction of Fig.7. The first three pictures are enhancements due to averaging with thresholding, filtering by using the Laplacian filter and removal of negative entries. The enhanced pictures are poor in quality. Only the edge preserving smoothing method gives a reasonably good enhanced picture as can be seen from Fig. 22.

In conclusion one can say that if a picture of good quality is required, it is better to increase the number of angles than to use an enhancement scheme. This is because the successful implementation of any enhancement scheme requires an accurate foreknowledge of the mathematical nature of the degradation. The degradation resulting while trying to reconstruct a picture with limited projections has not yet been mathematically characterized.

Fig.5.1

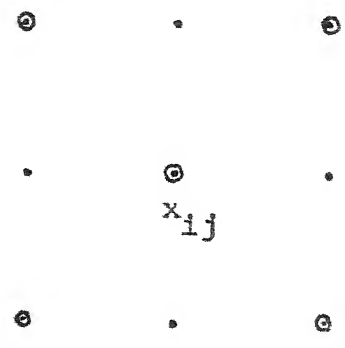
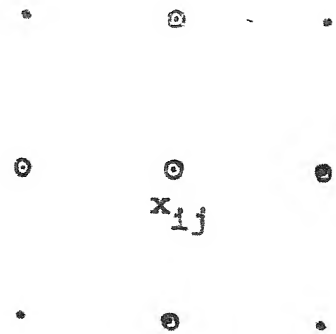


Fig.5.2



4 point averages; the points to be considered for the centre point (x_{ij}).

Fig.23-26 show the result of simulations with noise. Fig.23 is the picture of uniformly distributed random noise between 0 and 1. The result of reconstructing this is shown in Fig.24. The reconstructed picture shows smoothing of noise due to the process of averaging. This noise is added to the original picture and the result is shown in Fig.25. Fig.26 is the reconstruction of the noisy picture. It is to be noted, that though it appears as if some lines in the dark portion of the picture have been removed, the fact is that the character substituting for these lines (0) has a intensity value greater than the characters of the background (Z,),X etc.). A better overprinting subroutine would surely have resulted in a more visually satisfying picture.

Fig.1.

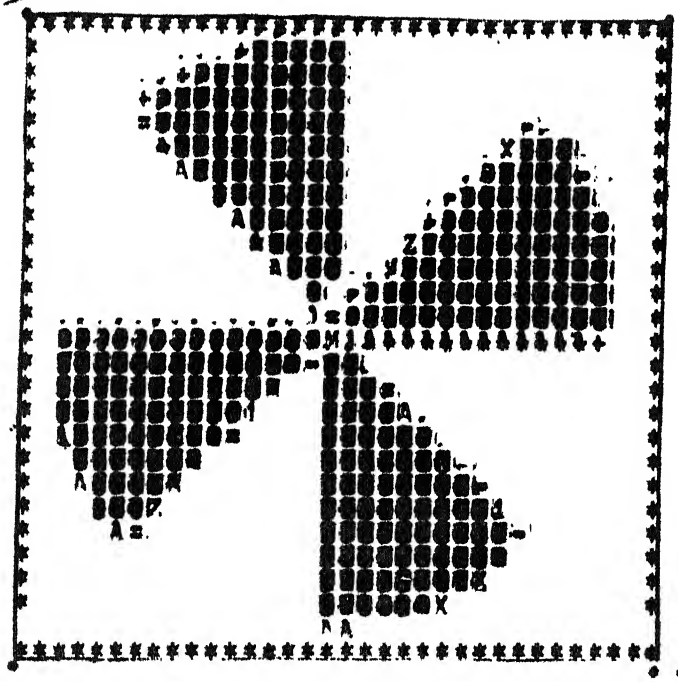
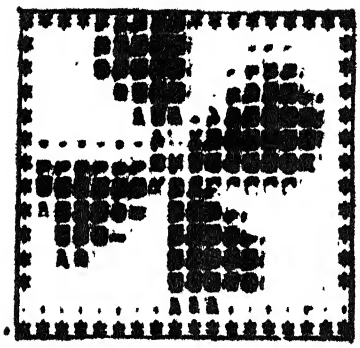


Fig.2.



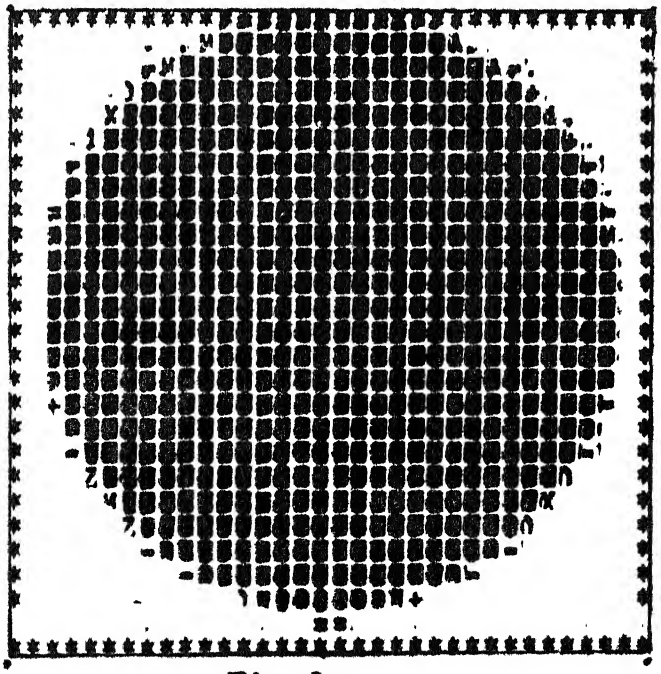


Fig.3.

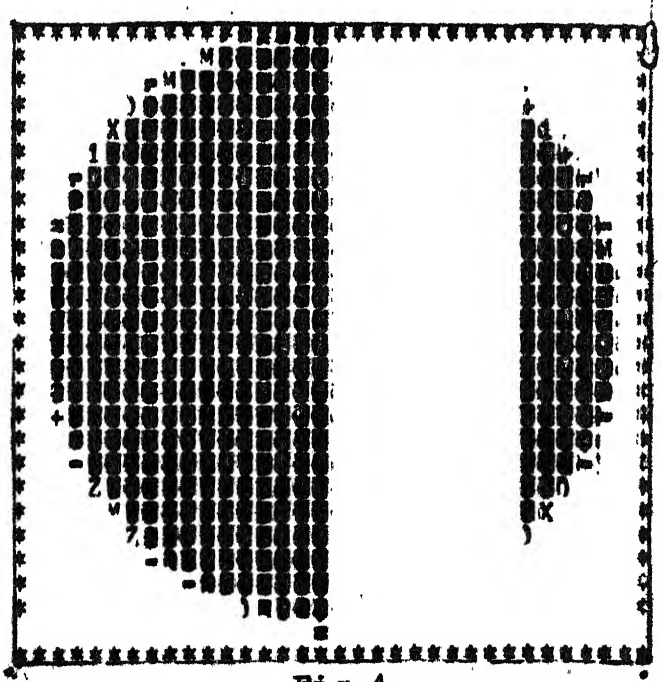


Fig.4.

Fig.5.

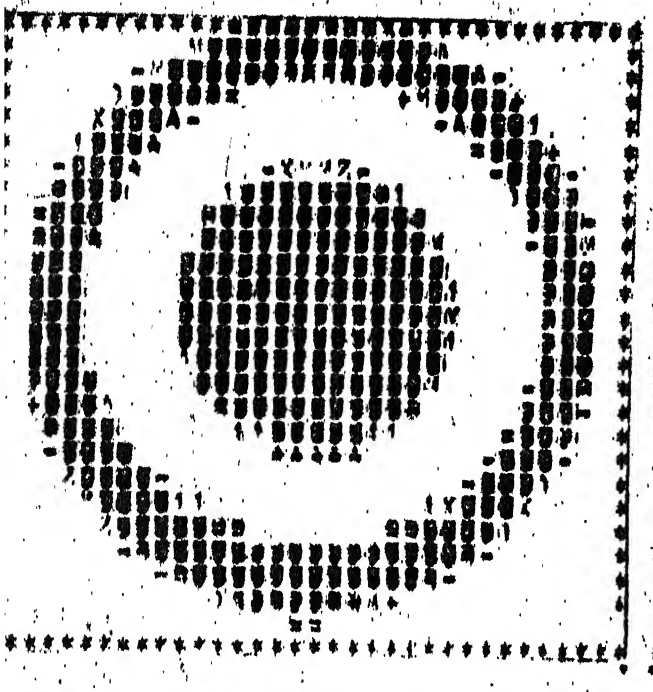
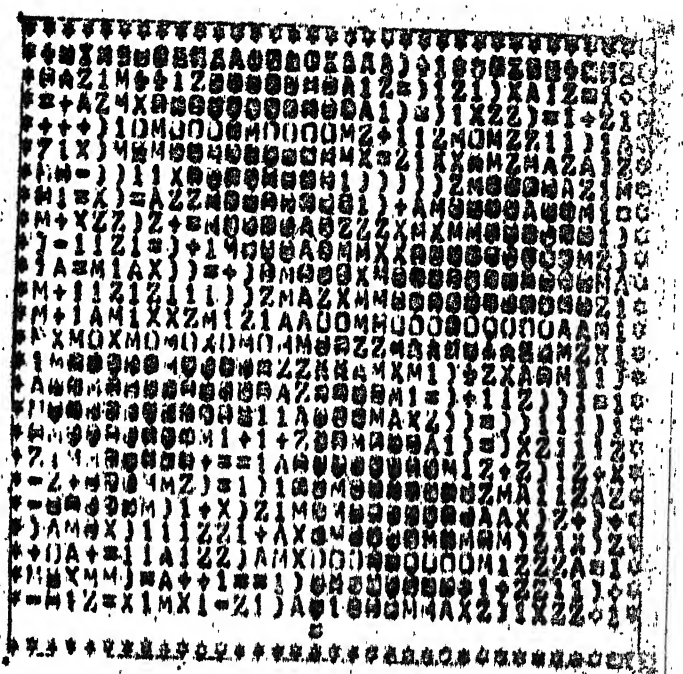


Fig.6.



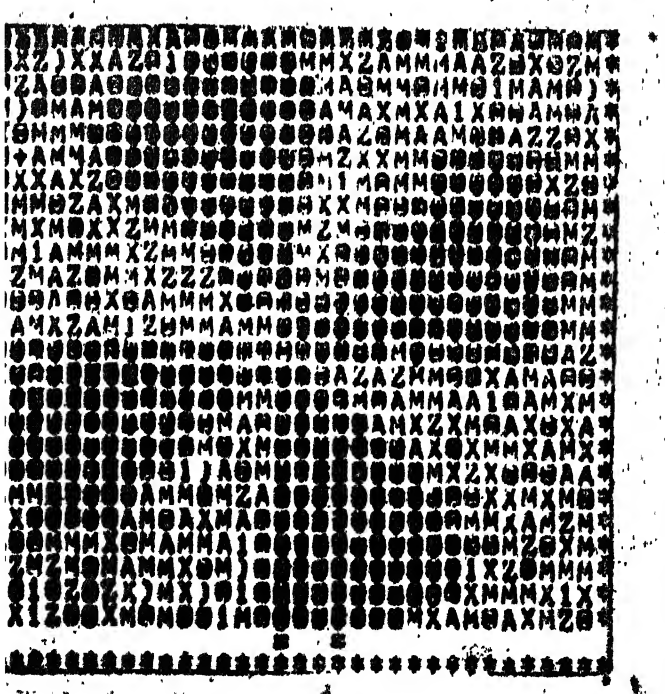


Fig.7.

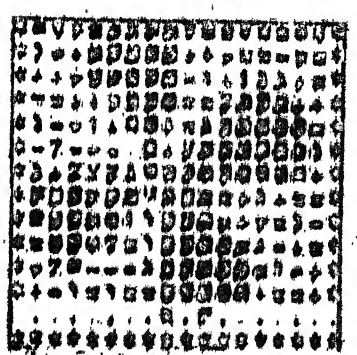
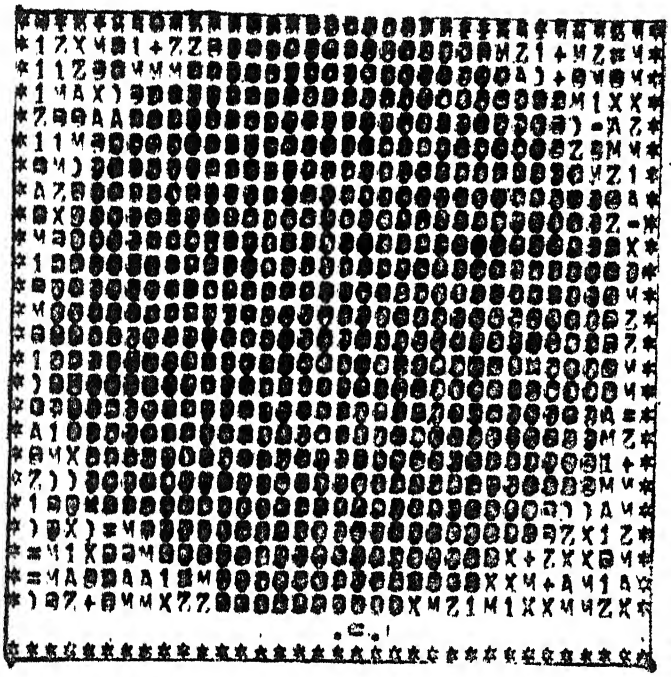
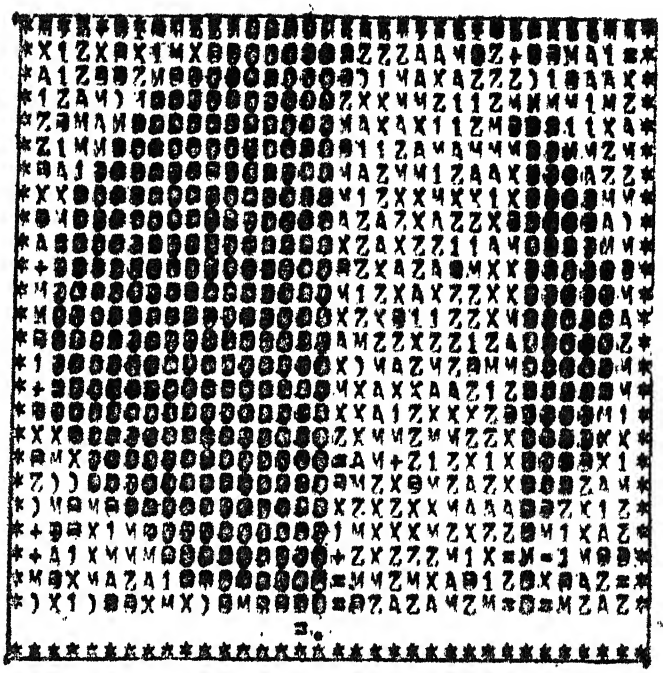


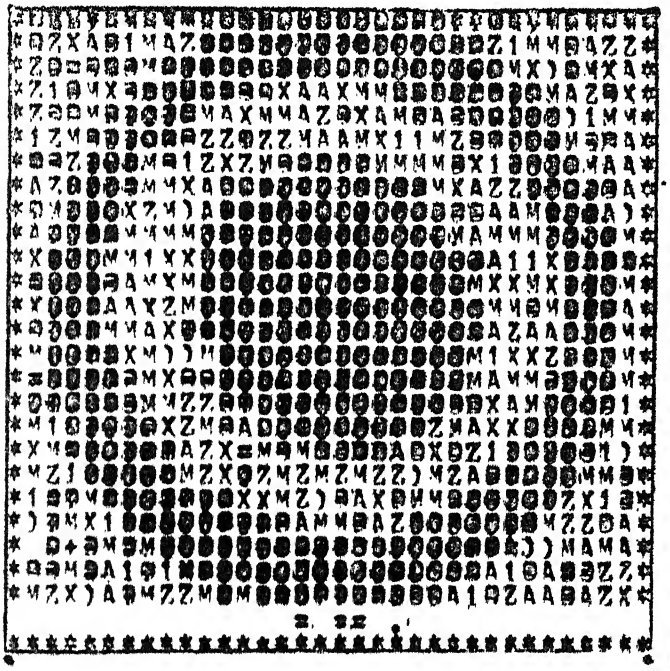
Fig.8.

Fig.9.

Fig.10.







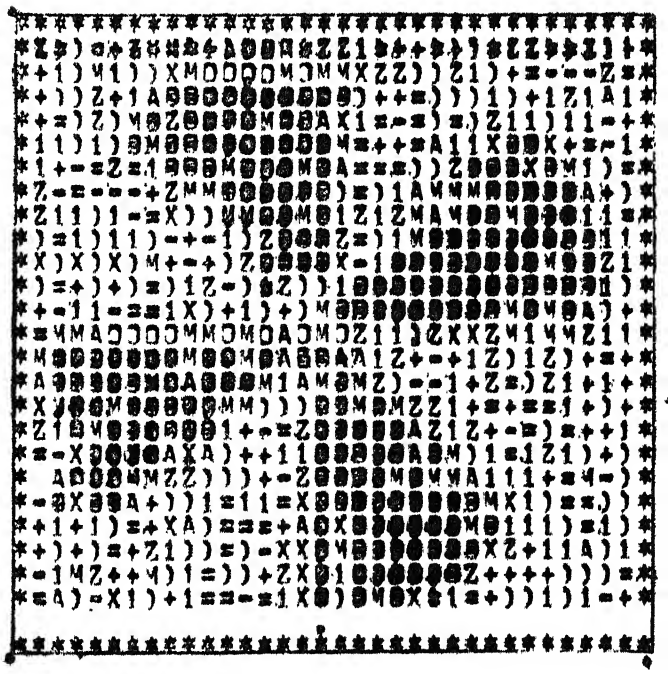


Fig.11.

Fig.12.

Fig.13.

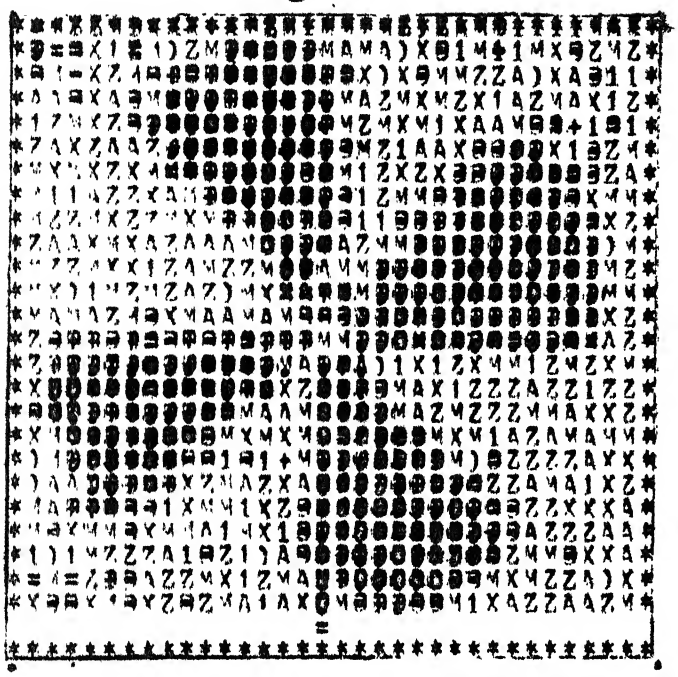
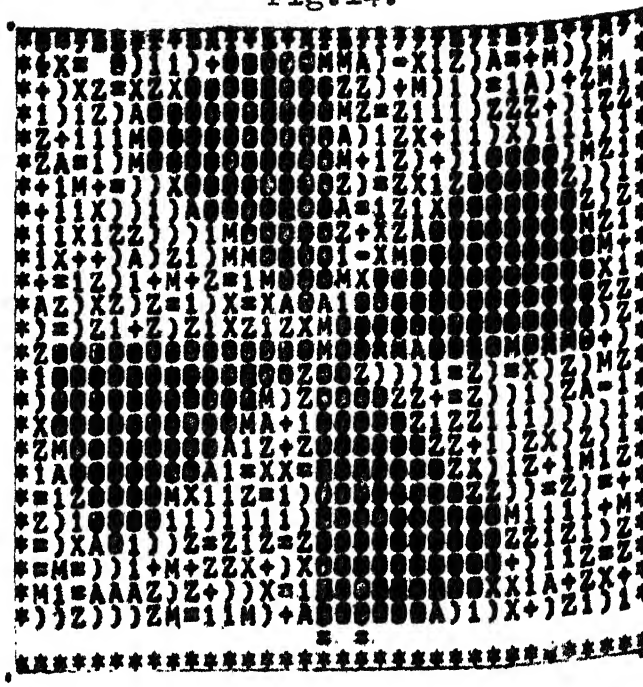


Fig.14.



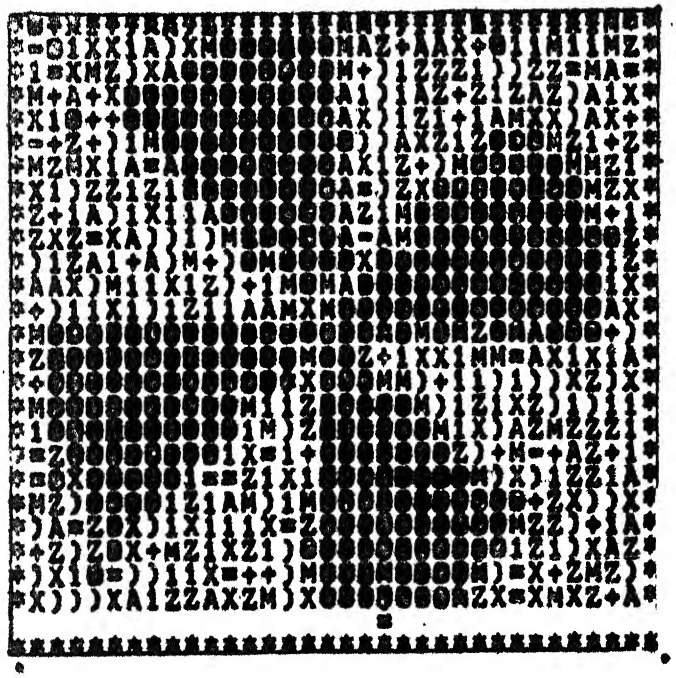


Fig.15.

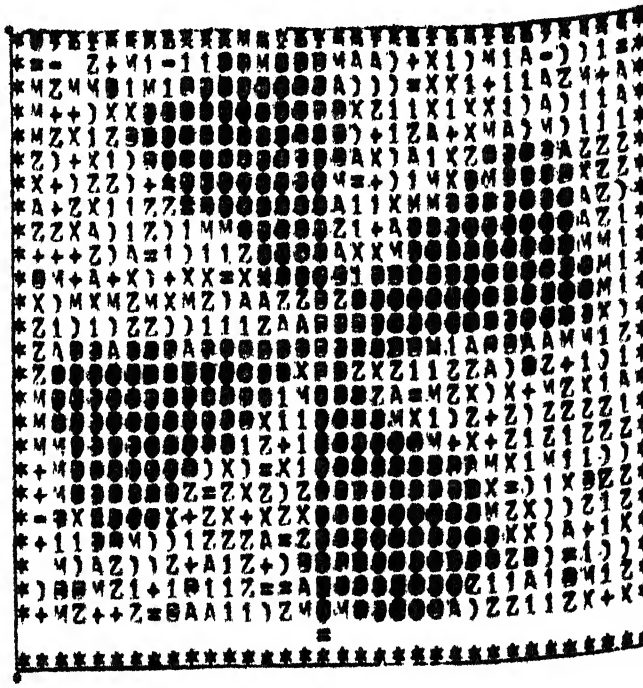


Fig.16.

Fig.17.

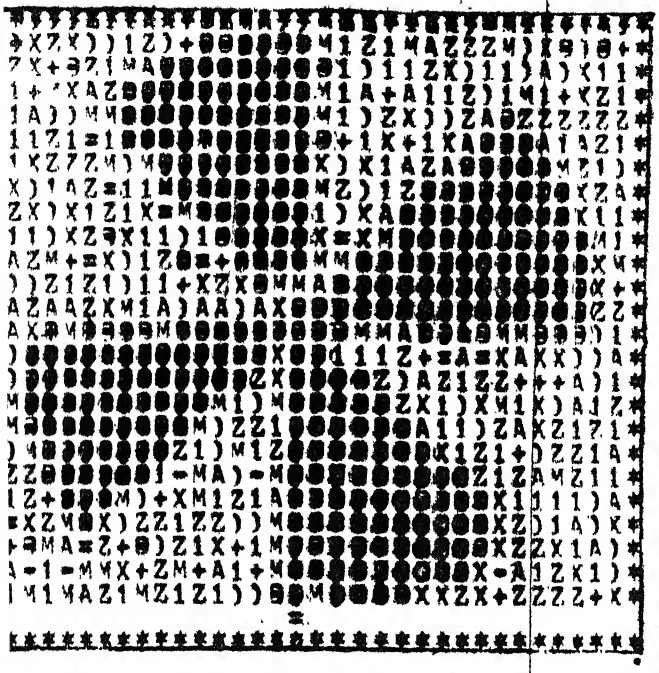
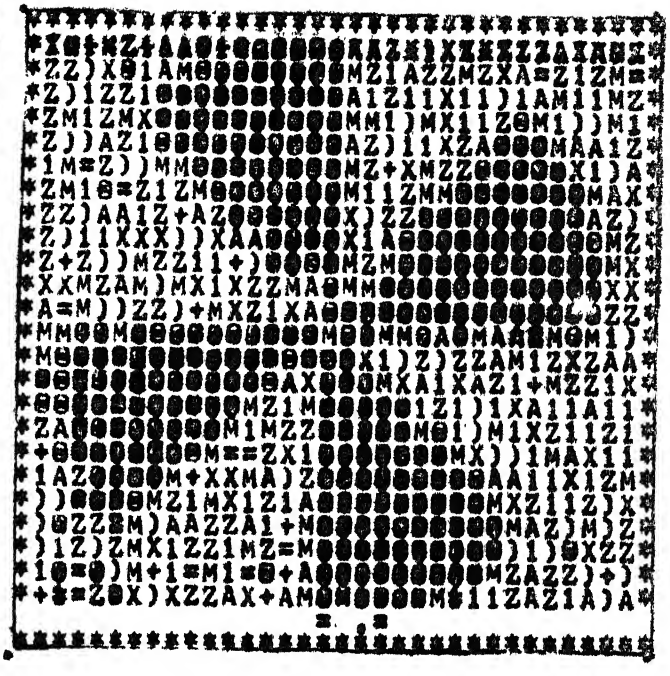


Fig.18;



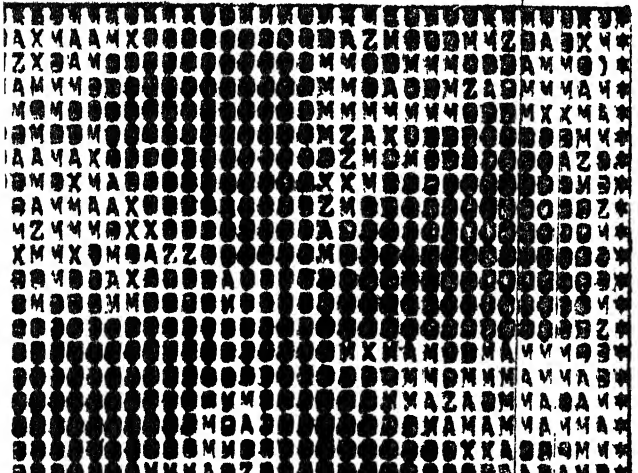


Fig.19.

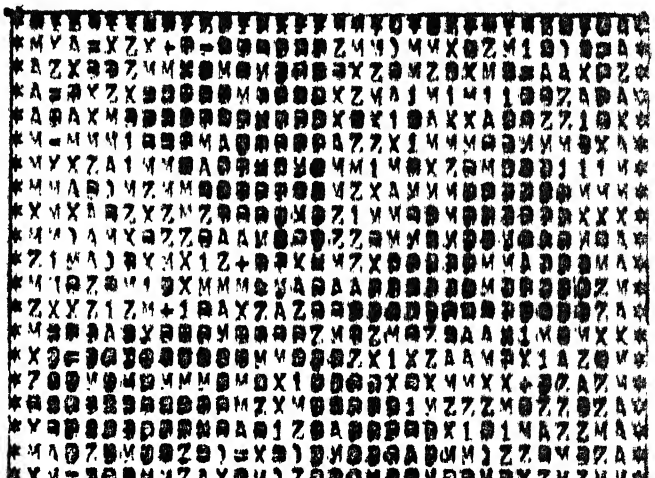


Fig.20.

Fig.21.

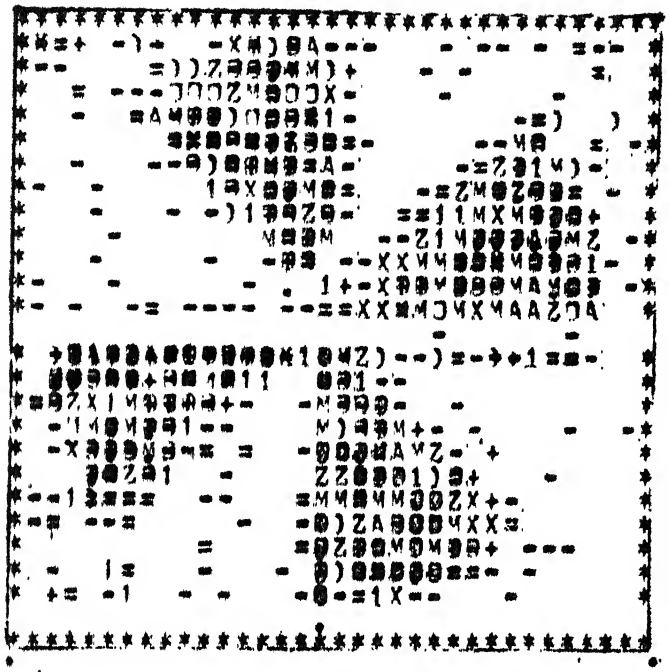
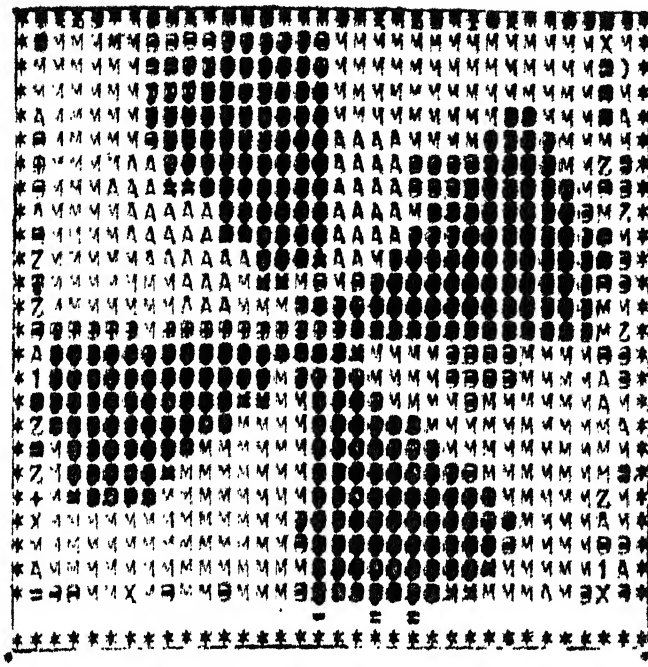


Fig.22.



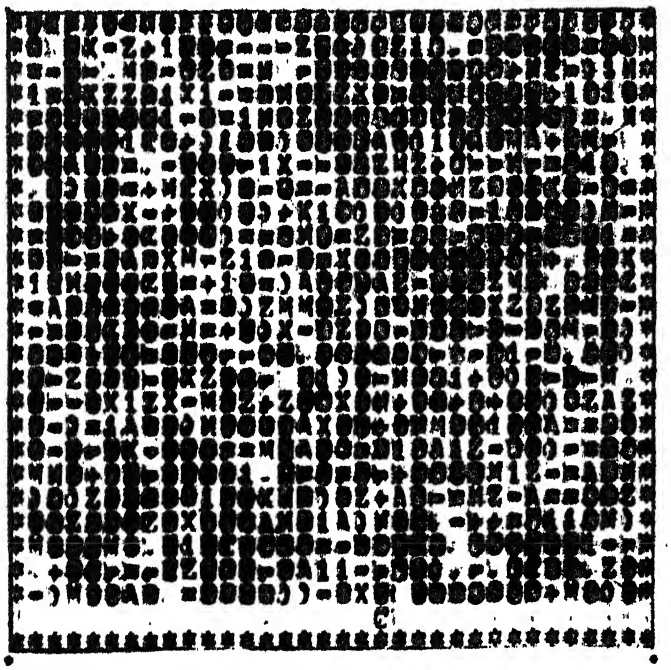


Fig.23.

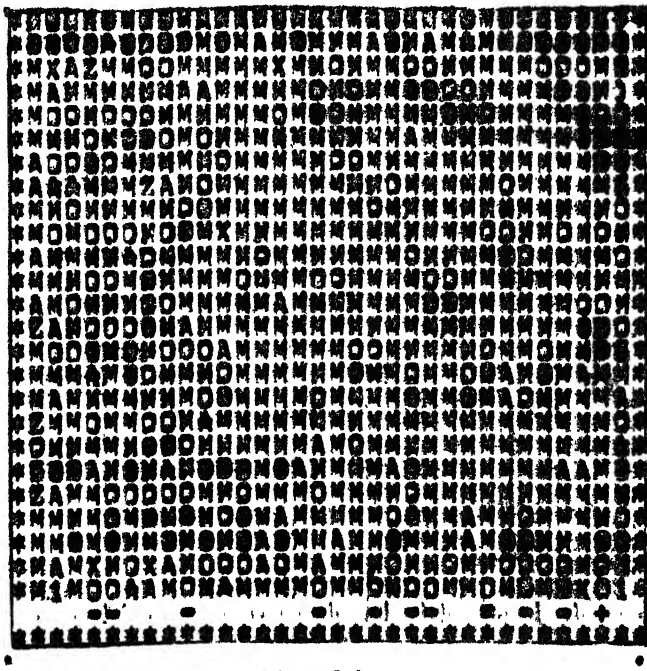


Fig.24.

Fig.25.

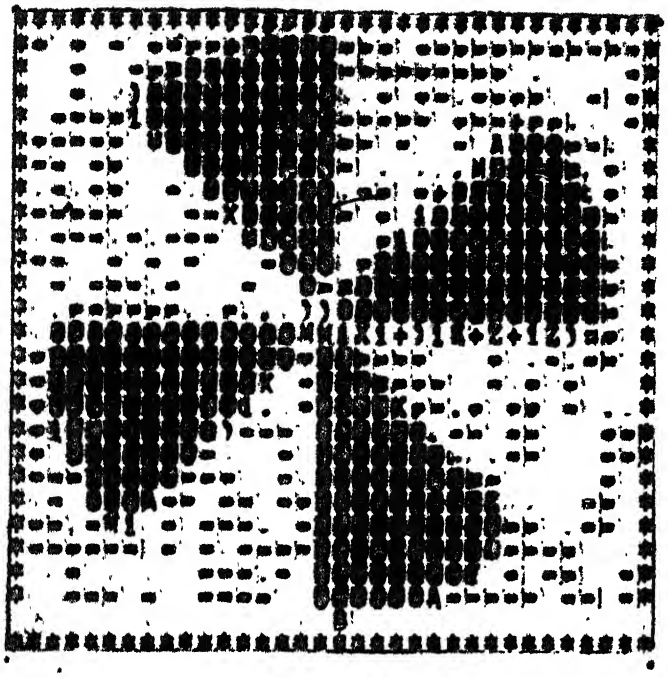
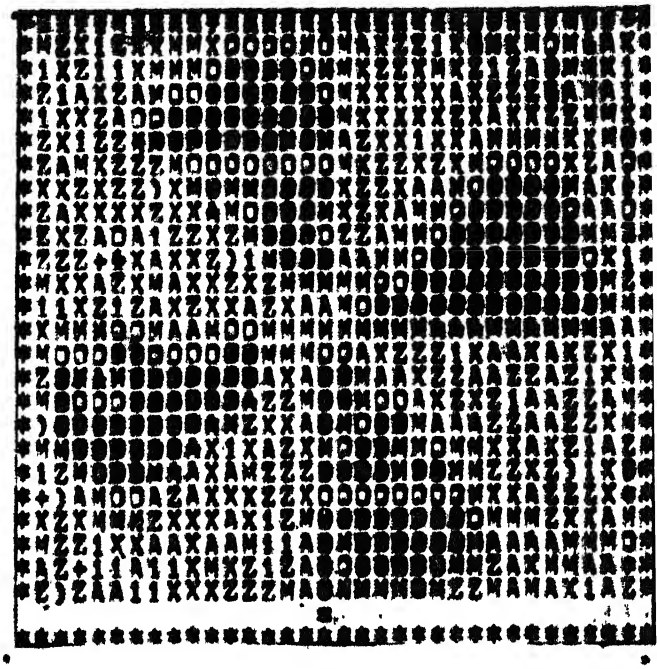


Fig.26.



CHAPTER 6

CONCLUSION

A powerful iterative method namely the Residual Projection Algorithm has been used to reconstruct pictures from their projections. Due consideration has always been given to the fact, that for practical reasons, the number of projections is always much less than the total number of pixels. This gave rise to poor reconstructions and the problem was solved by adding physical constraints in the form of averaging and median filtering. The resulting picture was good in quality.

The computational efforts for each of the various popular methods have been calculated and it has been shown that our method compares favourably in this respect too. It requires the same order of computations as ART and is free from operations like back projection. Back projection, it may be mentioned, is the step that slows down the otherwise fast Fourier Reconstruction methods. An additional useful feature of this method is its ability to reconstruct in the presence of noise.

Enhancement schemes have been tried, with very little success, to improve the picture quality. The degradation of a reconstructed picture has yet to be mathematically

characterized. Until this is done, it will be very difficult to find a suitable enhancement scheme to improve the quality of the reconstructed picture. Till then, the only way of obtaining a better picture will be to reconstruct it with an increase in the number of angles and hence the number of equations.

Based on our experiments we suggest any one of the two methods. 1) Residual projection with averaging, preferably the 4 point average of Fig.5.2, at each iteration, 2) Residual projection with median filtering at each iteration. Both of these methods yield satisfactory results.

Here, then, we have a method which, because of its block iterative nature, is useful in handling large systems, is reasonably fast and results in reconstructions which are favourably comparable with reconstructions from other algorithms. In fact, it satisfies all the requirements of a 'good' algorithm.

REFERENCES

1. Andrews, H.C., Hunt, B.R., Digital Image Restoration, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1977.

2. Nagao, M., Matsugama, T., Edge Preserving Smoothing, Proceedings of the Fourth International Joint Conference on Pattern Recognition, 1978.

3. Rosenfeld, A., Kak, A.C., Digital Picture Processing, Academic Press, Inc., New York,1976.

4. Rabiner, L.R., Gold, B., Theory and Application of Digital Signal Processing, Prentice Hall, Englewood Cliffs, New Jersey, 1975.

5. Tanabe, K., Projection Method for solving a singular system of linear equations and applications, Numerische Mathematik, 17, pp. 203-214, 1971.

6. Rathore,R.K.S., TheMethod of Residual Projections, Sent to Numerische Mathematik for publication.

7. Conte, S.D., de Boor, C., Elementary Numerical Analysis, (an algorithmic approach), McGraw-Hill, Kogakusha, Ltd., 1972.

8. Rao, C.R., Mitra, S.K., Generalized Inverse of Matrices and its Applications, John Wiley and Sons, Inc., New York, 1971.

9. Cho, Z.H., General Views on 3-D Image Reconstruction and computerized Axial Tomography, IEEE Trans. on Nuc. Sc., Vol.NS-21, June 1974, pp. 44-71.

10. Oppenheim, B.E., More Accurate Algorithms for Iterative 3-D Reconstruction, IEEE Trans. on Nuc.Sc., Vol.NS-21, June 1974, pp. 72-77.

11. Gordon, R., Herman, G.T., Bender, R., ART for 3-D Electron Microscopy and X-ray Photography', Journal of Theoretical Biology, 29, pp. 471-481, 1970.

12. Krishnamurthy, E.V., Sen, S.K., Computer Based Numerical Algorithms, Affiliated East-West Press Pvt. Ltd., New Delhi 1976.

13. Huesman, R.H., Gullberg, G.T., Greenberg, W.L., Budinger, T.F., Donner Algorithms for Reconstruction Tomography, Lawrence Berkeley Laboratory, University of California, Berkeley.

14. Ramachandran, G.N., and Lakshminarayan, A.V., Three Dimensional Reconstruction from Radiographs: Application of Convolution Methods Instead of Fourier Transforms, Proc. Nat. Acad. Sc. U.S., 68(9), pp. 2236-2240, 1971.

15. Gordon, R., A. Tutorial on ART, IEEE Trans. on Nuc.Sc., NS 24(3), pp. 78-93, 1974.

16. Oppenheim, B.E., More Accurate Algorithms for Iterative 3-D Reconstruction, IEEE Trans. on Nuc.Sc. NS-21, pp.72-77, 1974.

17. Singh,S., Reconstruction from Projections - A Comparative Study, M.Tech. Thesis (I.I.T.Kanpur, India).

18. Ramakrishna, R.S., Some Iterative Techniques in Digital Image Restoration, Ph.D. Thesis (I.I.T.Kanpur, India).

19. Kashyap, R.L., Mittal, M.C. in Proc. First Int. Joint Conf. Pattern Recognition, Washington, pp. 286-292, 1973.

20. Hunt, B.R., Digital Image Processing, Proceedings of IEEE, Vol.63, No.4, pp. 693-708, April 1975.

21. Cappellini, V., Constantinides, A.G., Emiliani, P., Digital Filters and their Applications, Academic Press, Inc. London, 1978.

22. Demidovich, B.P., Maron, I.A., Computational Mathematics, Mir Publishers, Moscow, 1976.

23. Gupta, L., Orthogonalization Strategy for Fast Reconstruction from Projections, M.Tech. Thesis (I.I.T.Kanpur, India).

24. Smith, K.T., Solomon, D.C., Wagner, S.L., Practical and Mathematical Aspects of the Problem of Reconstructing Objects from Radiographs, Bull. of the Amer.Mat.Soc., Vol.83, No .6, Nov.1977.

25. Special Issue on Image Restoration and Enhancement, IEE Proceedings, Vol.127, Part E, No.5, Sept.1980.

26. Justusson, B.P., Noise Reduction by Median Filtering, Proceedings of the Fourth International Joint Conference on Pattern Recognition, 1978.

27. Ben-Israel, A., Greville, T.N.E., Generalized Inverses, Theory and Applications, John Wiley & Sons, Inc., New York.

28. Zuhair Nashed, M., Generalized Inverses and Applications, Academic Press, New York, 1976.

29. Micchelli, C.A., Rivlin, T.J., International symposium on Optimal Estimation in Approximation Theory, Freudenstadt, Germany, 1976, Plenum Press, New York, 1977.

30. Budinger, T.F., Gullberg, G.T., Three Dimensional Reconstruction in Nuclear Medicine Emission Imaging, IEEE Trans. on Nuc. Sc. Vol.NS-21, June 1974, pp. 2-20.

31. Shepp,L.A., Logan, B.F., The Fourier Reconstruction of a Head Section, IEEE Trans. on Nuc.Sc., Vol.NS-21, June 1974, pp. 21-43.

32. Gilbert, P.F.C., Iterative Methods for Reconstruction of 3-D objects from Projections, J.Theo.Bio., 36, pp. 105-117, 1972.

33. Gordon, R., Herman, G.T., Three Dimensional Reconstruction from Projections: A Review of Algorithms, Proc.of the Soc.of Photo-optical Instrumentation Engineers, Vol.47, 1975.

34. Wee, W.G., Hsieh, T., An Application of the Projection Transform Technique in Image Transmission, IEEE Transactions on Systems, Man and Cybernetics, Vol.SMC-6, No.7, July 1976.

35. Kwoh, Y.S., Reed, I.S., Truong, T.K., A Generalized $|w|$ - Filter for 3-D Reconstruction, IEEE Trans.on Nuc.Sc., Vol.NS-24, No.5, October 1977.

36. Herman, G.T., Lent, A., Lutz, P.H. Relaxation Methods for Image Reconstruction, CACM, Vol.21, No.2, Feb. 1978.

37. Davies L.S., Rosenfeld, A., Noise Cleaning by Iterated Local Averaging, IEEE Trans.on Sys., Man, Cyber., Vol. SMC-8,No.9, Sept.1978.